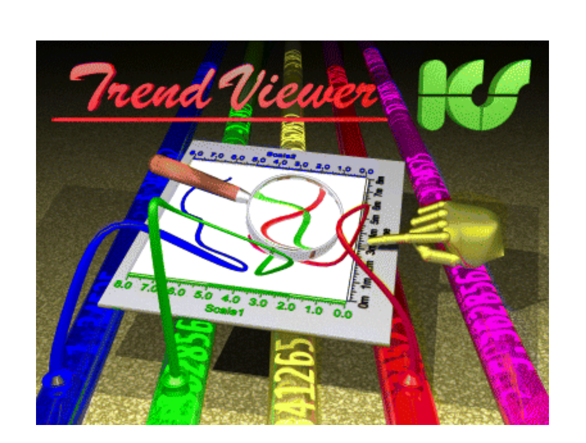


TrendViewer



.NET 2.0 Komponente zur Darstellung und Analyse von Messkurven

Benutzerhandbuch

Version 4



TrendViewer Lizenzbedingungen

.NET 2.0 Komponente zur Darstellung und Analyse von Messkurven

Für die Benutzung gelten die folgenden Lizenzbedingungen:

1. Mit Annahme dieser Lizenzbedingungen erwirbt der Lizenznehmer ein zeitlich unbegrenztes, örtlich nicht beschränktes einfaches nichtausschließliches Nutzungsrecht dieses Produktes. Die Nutzung dieses Produktes zur Entwicklung von Applikationen ist nur dem Lizenznehmer gestattet. Die Lizenz ist für einen Arbeitsplatz gültig. Die Weitergabe dieses Produktes als Bestandteil einer vom Lizenznehmer entwickelten Applikation ist nicht beschränkt.

2. Die in diesem Produkt enthaltenen Programme und sonstigen Dateien dürfen weder im Quelltext noch als ausführbare Programme - auch nicht in veränderter Form - zur Entwicklung von Applikationen an Dritte weitergegeben, verliehen oder gleichzeitig auf mehreren Computern eingesetzt werden. Auch die Installation einundderselben Lizenz auf mehr als einem Gerät oder mehrfache Nutzung im Netz - zum Beispiel in Firmen, Schulen oder Universitäten - ist untersagt.

3. Für *TrendViewer* wird die Entwicklerlizenz in Form einer Textdatei geliefert. Sie hat den Namen trv3net2developer.lic und darf keinesfalls mit der entwickelten Applikation weitergegeben werden. Mit der Applikation, in die *TrendViewer* integriert ist, muss keine Lizenzdatei ausgeliefert werden.

Wird *TrendViewer* bei der Projektierung von Prozeßbildern in Leitsystemen eingesetzt, muß nur auf dem Editor-Arbeitsplatz die Lizenzdatei trv3net2developer.lic installiert sein.

4. Alle weitergehenden Rechte bleiben dem Copyright-Inhaber vorbehalten. Die gelieferte Software darf weder disassembliert, dekompiliert oder anderweitig rückübersetzt werden.

5. Änderungen an der Software und Dokumentation bleiben ohne Mitteilungspflicht vorbehalten. Update-Service und Support werden nur registrierten Lizenznehmern gewährt.

6. Programm und Dokumentation wurden sorgfältig erstellt und geprüft. Außer im Falle grober Fahrlässigkeit oder Vorsatz übernimmt die ICS GmbH keine Haftung für etwaige Schäden, die auf Verwendung der Software oder der Dokumentation zurückzuführen sind. Desweiteren wird, sofern nicht ausdrücklich schriftlich zugesichert, keine Gewähr für die Verträglichkeit dieser Software mit irgendwelchen anderen Programmen oder Hardware-Bestandteilen übernommen.

7. Die Lizenzbedingungen erstrecken sich ebenfalls - auch ohne erneute Mitteilung - auf alle eventuellen Updates oder Ergänzungen.

8. Sollte einer der vorangegangenen Punkte ungültig werden, bleiben die restlichen davon unberührt.

Alle im Text verwendeten Warenzeichen werden anerkannt.



Inhalt















1. Einleitung

Anwendungsbereich

TrendViewer ist eine .NET Komponente (basiert auf dem Microsoft .NET Framework ab Release 2.0) mit hoher Usability, die in Applikationen zur Darstellung und visuellen Analyse von Messdaten integriert wird. TrendViewer lässt sich auch in die Benutzeroberflächen von Leitsystemen, Messwerterfassung- und Analyseprogrammen oder in Maschinensteuerungen integrieren, die die Verwendung von .NET Komponentes unterstützen.

TrendViewer bietet ein sogenanntes "Zoomable User Interface" an und zeichnet sich insbesondere durch die hochflexible, exakte Darstellung von Skalen unterschiedlicher Typen aus. Die große Anzahl von Kurven und Skalen, die TrendViewer gleichzeitig bearbeitet, setzen den Anforderungen nach der Bearbeitung sehr großer Messdatenmengen nahezu keine Grenzen. Die zahlreichen zusätzlichen Objekte und interaktiven Eigenschaften ermöglichen es dem Benutzer, auch in sehr komplexen Kurvendiagrammen die Übersicht zu behalten und die gewünschten Daten schnell und direkt darstellen zu können. Weitere Funktionen unterstützen die Applikation bei der visuellen Auswertung großer Datenmengen.

TrendViewer ist vor allem dazu geeignet, aus umfangreichen Messkurven, wie sie z.B. in der Prozessindustrie (Pharmazie, Chemie) aus Gründen der Qualitätskontrolle aufgezeichnet werden, Ausschnitte interaktiv mit der Maus einzustellen. Der Ausschnitt wird in seinen Ausmaßen und seinem Verhältnis zur gesamten Messdatenmenge während der Mausbewegung unmittelbar dargestellt. Dabei ist TrendViewer transparent bezüglich der Datenquelle. Online-Trendanzeigen sowie Archive lassen sich gleichermaßen einfach mit TrendViewer bearbeiten.

Zwei Aufgabenstellungen werden von TrendViewer besonders unterstützt:

- die Darstellung von Zusammenhängen zwischen Messkurven über die Zeit in sogenannten Phasendiagrammen. Durch das interaktive Positionieren von Skalen lassen sich die Zusammenhänge leicht neu definieren
- der direkte Vergleich von Messkurven einer Messstelle in unterschiedlichen Zeitintervallen. Die Zeitintervalle lassen sich leicht mit der Maus verändern und mit den 2 x-Skalen unabhängig voneinander verschieben.

Lizenzverfahren

Die Komponente wird Software-Entwicklern oder Projektierern zunächst als lizenzfreie Testversion angeboten, damit die Eigenschaften de Komponente im Zusammenhang mit einer geplanten Anwendung ausprobiert werden können. Die Funktionalität der lizenzfreien Version ist nicht eingeschränkt.

Wenn TrendViewer in einer Anwendung an Endkunden weitergegeben werden soll, ist eine Entwicklerlizenz zu erwerben. Andernfalls erscheint während der Benutzung der Anwendung ein Hinweis darauf, dass eine unlizenzierte Version von TrendViewer verwendet wurde. Zur Laufzeit ist keine weitere Lizenz erforderlich.



Installation

TrendViewer wird über das Internet zum Herunterladen und auf Datenträger (CD) angeboten. Die Installationsdatei enthält die Trend Komponente und ein Testprogramm, mit dem die Eigenschaften des TrendViewer eingestellt und mit der vorherigen Version dieses Produktes verglichen werden können. Dokumentation sowie Programme im Sourcecode zur Demonstration der Integration des TrendViewer in Applikationen werden ebenfalls bereitgestellt.

Anforderungen an Computer und Software

Da TrendViewer eine vor allem bei der interaktiven Benutzung sehr dynamische Komponente ist, sollte der Computer, auf dem die fertige Applikation läuft, abhängig vom eingesetzten Betriebssystem folgende Leistungsdaten nicht unterschreiten, damit TrendViewer ergonomisch genutzt werden kann:

Personal Computer mit Pentium V 1 GHz, 512 MByte RAM, Maus mit 2 oder 3 Tasten oder ein ähnliches Zeigegerät.

Wenn TrendViewer in der Benutzerschnittstelle eines Leitsystems verwendet wird, sind die Anforderungen, die der Hersteller des Leitsystems angibt, zu beachten.

Folgende Software ist erforderlich:

Microsoft Windows NT 4.0, Windows 2000, Windows XP, Windows Vista oder neuer.

Um TrendViewer in eine Applikation zu integrieren wird eine Entwicklungsumgebung wie z.B. Microsoft Visual Studio benötigt. Bei der Integration in Prozessbilder sind die Editoren der jeweiligen Leitsysteme zu verwenden.

Lizenzierung

Die Lizenzierung wird über das Internet vorgenommen. Zur Erzeugung einer Lizenzdatei sind der Name einer Person, der Firmen- und Abteilungsname sowie der Standort erforderlich. Diese Daten können über das Formular auf dem Web Site www.icsgmbh.de oder den eShop übermittelt werden. Der Käufer erhält per E-Mail die personalisierte Lizenzdatei.

Für TrendViewer wird die Entwicklerlizenz in Form einer Textdatei geliefert. Sie hat den Namen trendviewer.lic und darf keinesfalls mit der entwickelten Applikation weitergegeben werden. Mit der Applikation, in die TrendViewer integriert ist, muss keine Lizenzdatei ausgeliefert werden.

Wird TrendViewer bei der Projektierung von Prozessbildern in Leitsystemen eingesetzt, muss nur auf dem Editor-Arbeitsplatz die Lizenzdatei trendviewer.lic installiert sein.



Produktarten

TrendViewer ist in 3 Leistungsstufen verfügbar, die in unterschiedlichen Lizenzdateien ausgegeben werden:

Produktart	Eigenschaften
Ohne Lizenz	.NET 2.0 Komponente zur Anzeige von bis zu 512 binären oder analogen Messkurven mit 512 y-Skalen, 2 x-Skalen, jeweils lineare und logarithmische Skalierung. X-Achsen für Zeit, Dauer, Winkel und Wert. Darstellung von Zeitkurven, Ortskurven und Phasendiagrammen. Einsetzbar unter Windows 2000, Windows XP und Windows Vista. Alle Funktionen, außer der in der Professional Edition enthaltenen Zusatzfunktionen, können ohne Lizenz getestet werden
Standard Edition	Tasten zum Ein- und Ausblenden der Kurven, Umlaufpuffer für über 500 binäre und analoge Messkurven, freie Zuordnung zu x- und y-Skalen, Hüllkurvendarstellung, 2 verstellbare Grenzwertlinien für jede Kurve, Fluten zwischen Kurven, Fluten zwischen Kurve und Grenzlinie, beliebige Linien- und Füllstile, 2 Zeitskalen für Meßdauer oder absolute Zeit zum interaktiven Kurvenvergleich, mehr als 500 lineare und logarithmische x- und y-Skalen, zyklische Grad-Skalen, wertbezogene äquidistante Hilfslinien pro Skala zur Unterstützung der qualitativen Ablesbarkeit, Übersichtsbalken zum Vergleich des Kurvenausschnitts mit dem Messintervall, Tooltip mit Wertangabe an der Mausposition, Dreistufige Skaleneinteilung, (y,t)-, (x,y)- und Phasendiagramme, senkrechte oder waagerechte Darstellung, automatische Anpassung an die Fenstergröße, Kurvenfeldgröße einstellbar, Streifendiagramme, Lasso Zoom, Follow me“ Tasten für Kurvenschreiberfunktion, Kurvenfangfunktion
Professional Edition	Die *TrendViewer* Professional Edition bietet die gleichen Funktionalitäten wie die *TrendViewer* Standard Edition. Darüber hinaus bietet die Professional Edition: verschiebbare Kurven, interaktiv editierbare Kurvenpunkte, interaktive Skalenpositionierung nebeneinander oder übereinander mit Zielerkennung, das Lineal Fenster



Support

Wenn bei der Benutzung von *TrendViewer* technische Unterstützung benötigt wird, sind die Fragen zusammen mit der Lizenzdatei per E-Mail an die Adresse support-trv@icsgmbh.de zu senden. Informationen über neue Releases und Patches werden auf der Web Site www.icsgmbh.de veröffentlicht.

Informationen über neue Versionen von TrendViewer oder über andere Produkte können bei der Bestellung oder über den Eintrag in einer der Maillisten auf der Website der ICS GmbH angefordert werden.

Anschrift:

ICS Gmbh
Pforzheimerstr. 202c
D76275 Ettlingen
Deutschland

Telefon: ++49/7243/7127-0
Telefax: ++49/7243/7127-29

www.icsgmbh.de



2. Diagramm

Mit *TrendViewer* können einfache aber auch sehr komplexe Darstellungen von Messkurven parametriert werden. In einem Diagramm können bis zu 512 Kurven für analoge und binäre Messstellen einzeln oder in Gruppen ebenso vielen y-Skalen zugeordnet werden.

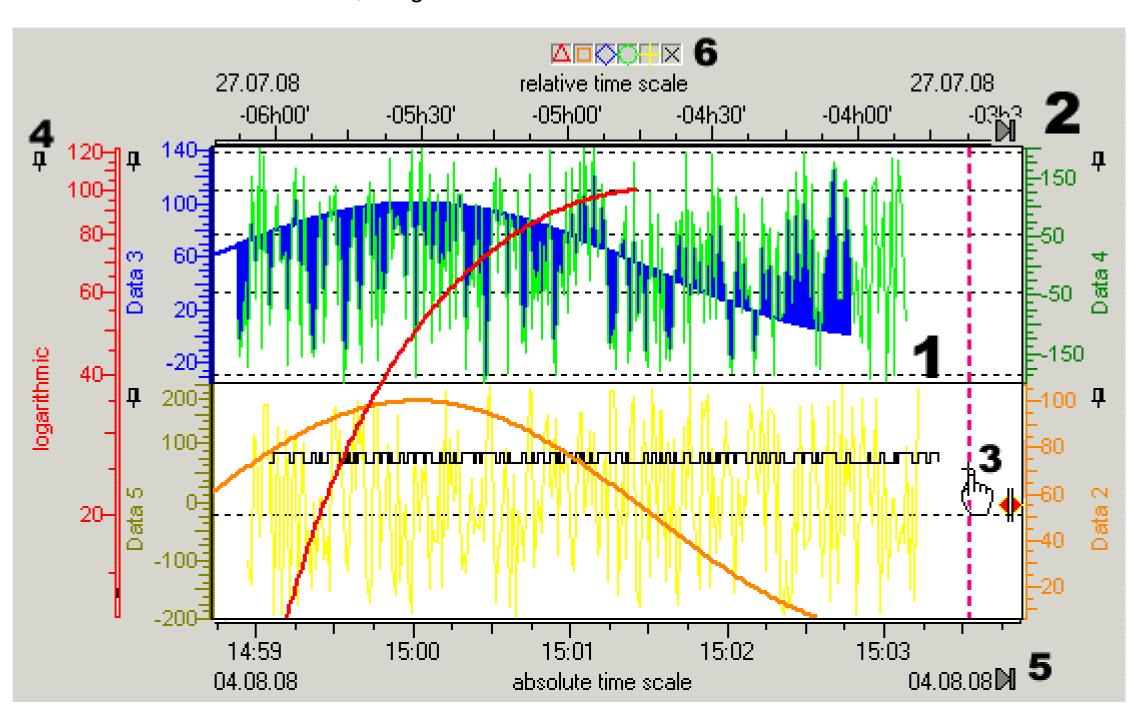
Da die Kurven einer der beiden voneinander unabhängigen x-Skalen zugeordnet werden können, lassen sich Kurvenverläufe von einer Messstelle in ein und demselben Diagramm zu unterschiedlichen Zeiten betrachten und vergleichen.

Eine besondere Art der Darstellung sind Phasendiagramme, in denen der Zusammenhang von Messwertkurven mehrerer Messstellen parallel zur Zeitachse wiedergegeben wird. Durch die interaktive Anordnung der Werteskalen können Bezüge zwischen den Messkurven direkt dargestellt werden.

Kurven können in Streifen übereinander im Trend-Fenster angeordnet oder über das ganze Trend-Fenster gezeichnet werden. Auch Kombinationen beider Darstellungen sind möglich, so dass komplexe Sachverhalte in einem Diagramm angezeigt werden können.

2.1 Komponenten

In der folgenden Abbildung sind alle Komponenten eines Diagramms, die *TrendViewer* an der Benutzerschnittstelle anbietet, dargestellt und bezeichnet.



Die Nummern in der Abbildung haben folgende Bedeutung:

1) Trend-Fenster mit 2 Streifen. Im oberen Streifen sind die Kurven zu den Skalen **Data 3** und **Data 4** dargestellt, im unteren Streifen die Kurven zu den Skalen **Data 2** und **Data 5**. Die Kurve zur Skala **logarithmic** läuft über das gesamte Trend-Fenster. Die Hilfslinien sind durch die logarithmische Skala vorgegeben. Außerdem ist eine Leselinie (3) eingeblendet, mit der *TrendViewer* Kurvenwerte an der jeweiligen Position an die Anwendung übergibt.



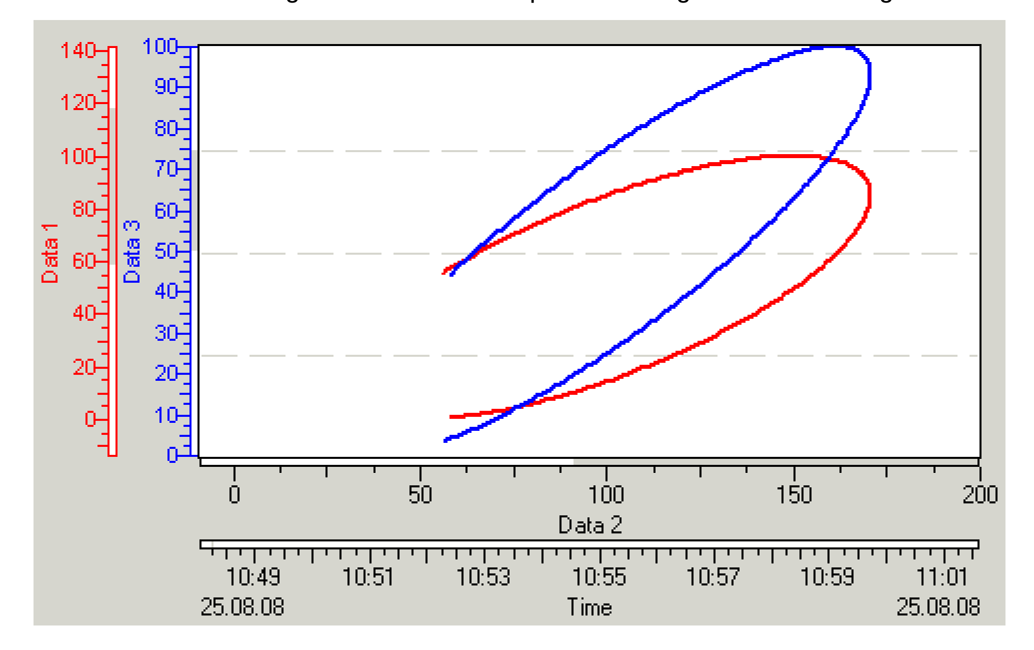
Im oberen Streifen sind 2 Kurven für analoge Messstellen zu sehen, die gegeneinander geflutet sind. Im unteren Streifen ist auch eine Kurve für eine binäre Messstelle (schwarz) zu sehen, der keine Skala zugeordnet ist.

Kurven lassen sich durch Kurvenknöpfe (6) ein- und ausblenden.

2) Um das Trend-Fenster herum sind Skalen angeordnet. In einem Zeitdiagramm sind z.B. oben und unten je eine Zeitskala und links und rechts Werteskalen platziert. Die Werteskalen können gestapelt sein. Skalen können interaktiv mit den Skalenknöpfen (4) an andere Positionen gezogen werden (Professional License erforderlich). Die "Follow-Me" Knöpfe (5) sorgen dafür, dass sich das Trend-Fenster wie ein Kurvenschreiber für aktuelle Daten verhält.

Die Komponenten und ihre Benutzung werden in den folgenden Kapiteln detailliert beschrieben.

Bei einem Phasendiagramm sind die Komponenten folgendermaßen angeordnet:



Die Kurven zu den Skalen **Data 1** und **Data 3** sind gegen die Kurve der Skala **Data 2** aufgetragen. Dadurch entsteht eine Ortskurve. Zusätzlich ist parallel zur Skala **Data 2** die Zeitachse zu sehen. Wenn das dargestellte Zeitintervall verringert oder erweitert wird, wird die Ortskurve entsprechend verkürzt oder verlängert. Wird das Zeitintervall verschoben, ist entsprechend ein anderer Ausschnitt der Ortskurve zu sehen.





2.2 Allgemeine Eigenschaften

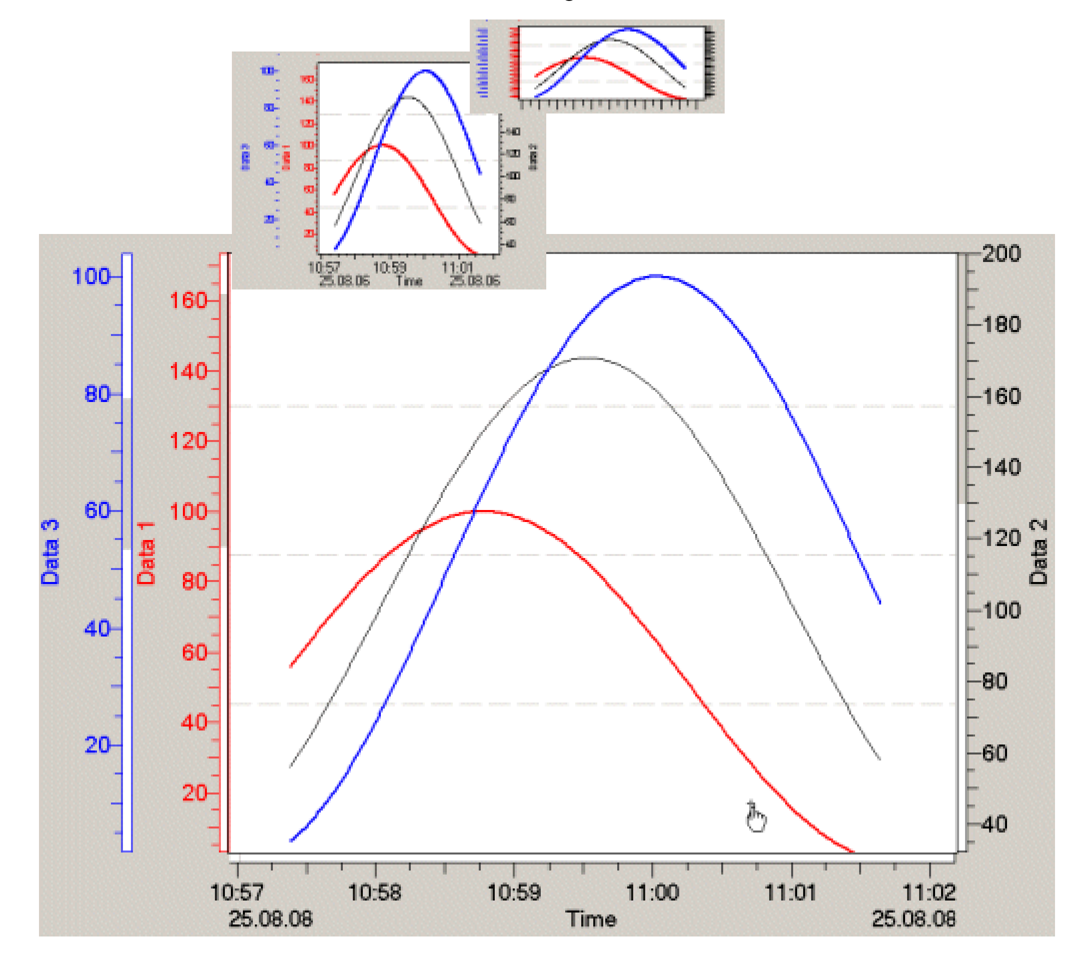
In der Fülle der angebotenen Möglichkeiten des *TrendViewer* gibt es eine Reihe von Eigenschaften, die das Aussehen und die Bedienbarkeit eines Diagramms übergreifend beeinflussen.

Hinweis: Bei den folgenden Beschreibungen von Komponenten des *TrendViewer* und ihrer interaktiven Bedienung wird immer die horizontale Ausrichtung des Diagramms beschrieben. Dabei verlaufen die x-Achsen waagerecht und die y-Achsen senkrecht. Bei Diagrammen in gedrehtem Zustand sind die Beschreibungen dementsprechend zu verstehen.

2.2.1 Größe des Diagramms verändern

Wenn *TrendViewer* in einer Windows-Anwendung verwendet wird, deren Größe veränderbar ist, so werden die Komponenten des Diagramms soweit als möglich in der Größe angepasst. Die Fontgröße der Skalenbeschriftung wird automatisch angepasst, wenn sie variabel eingestellt wurde. In extremen Fällen können z.B. die Beschriftung der Skalen oder auch ganze Teile des Trend-Fensters - wenn die Größe direkt vorgegeben wurde - wegfallen.

Unterschiedliche Größen desselben Diagramms:





2.2.2 Größe und Position des Trend-Fensters einstellen

Es gibt zwei Methoden, die Größe des Trend-Fensters festzulegen:

indirekt

Zunächst wird die Größe des Platzes für die Skalen durch Anordnung, Beschriftungsbreite, Schriftgröße und Zusatzelemente wie Übersichtsbalken oder Kurvenknöpfe errechnet. Der verbleibende Platz steht dem Trend-Fenster zur Verfügung.

direkt

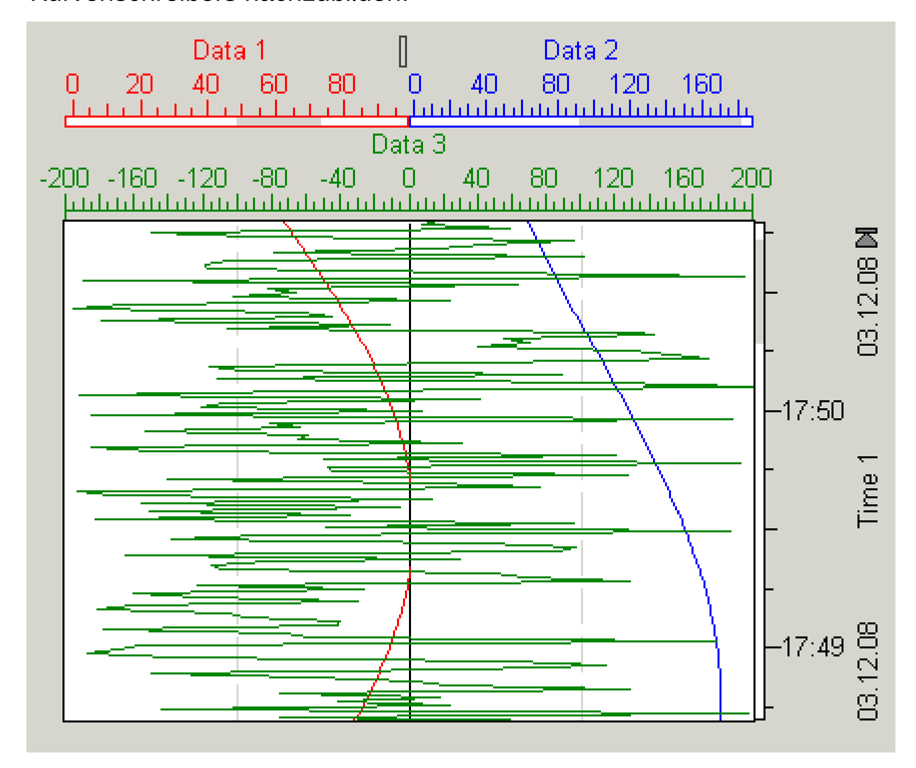
Der Abstand des Trend-Fensters vom Rand der gesamten zur Verfügung stehenden Fläche der *TrendViewer* Komponente wird vorgegeben. Dadurch wird der verbleibende Platz unter den Skalen aufgeteilt. Es hängt von den gewählten Parametern für die Skalenbeschriftung und die Verwendung der übrigen Skalenelemente ab, ob das entstehende Diagramm für den Benutzer gut lesbar ist.

2.2.3 Orientierung des Diagramms einstellen

Die Orientierung des Diagramms kann senkrecht oder waagerecht eingestellt werden. Die Beschriftung der Skalen ist festgelegt und richtet sich nach der Orientierung des gesamten Diagramms.

Durch zusätzliche Einstellungen an den Skalen kann der Verlauf der Messkurven von rechts nach links, von links nach rechts oder von oben nach unten eingestellt werden.

Ein Diagramm kann um 90° gedreht angezeigt werden, um das Aussehen eines mechanischen Kurvenschreibers nachzubilden:



Die Kurven zu den Skalen **Data 1** und **Data 2** werden im linken bzw. rechten Streifen gezeichnet, während die Kurve zur Skala **Data 3** über die gesamte Breite des Trend-Fensters läuft. Mit dem 'Follow-Me' Knopf werden die Kurven gestoppt bzw. laufen von oben nach unten, sobald neue Daten eintreffen.





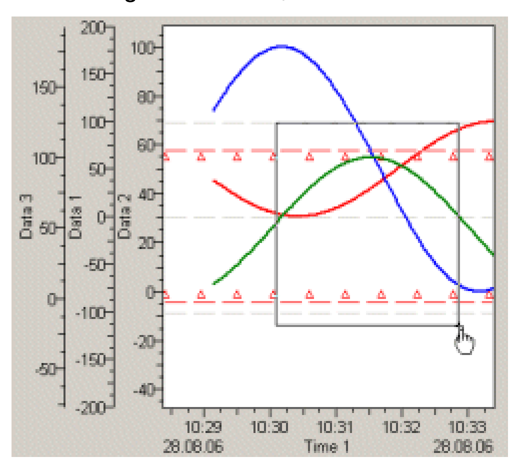
2.2.4 Zoomfunktion einstellen

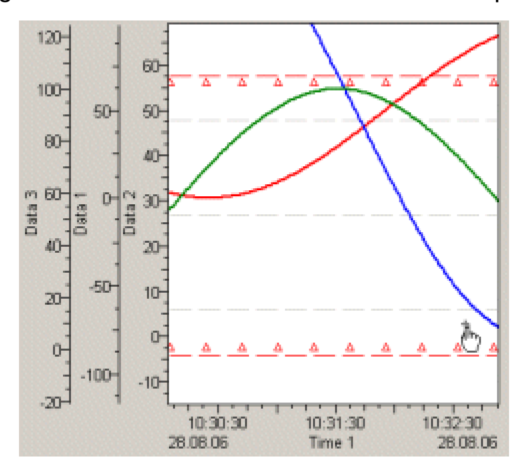
Wie ein Ausschnitt aus Messkurven eingestellt wird, ist im Kapitel 2.3.1 beschrieben. Die Zoom-Funktion ist abschaltbar, um die eingestellten Positionen vor versehentlichen Änderungen zu schützen.

Alternativ kann auch die Lasso-Technik eingesetzt werden, um einen Ausschnitt aus den Messkurven, die im Trend-Fenster zu sehen sind, einzustellen. Auch bei dieser Zoom-Technik werden die Zeit- und Werteskalen konsistent aktualisiert.

Der mit dem Lasso eingestellte Ausschnitt wird auf das ganze Trend-Fenster ausgedehnt. Dadurch sind mehr Details der Messkurven zu erkennen. Die Zoomfunktion kann mehrmals hintereinander angewendet werden. Die Zoom-Schritte sind allerdings nicht mit einer Undo-Funktion wieder rückgängig zu machen. Stattdessen muss ein von der Anwendung vor dem Zoomen gespeicherter Bereich wieder eingestellt werden.

Die Wirkung des Lasso-Zooms ist in der folgenden Abbildung zu sehen. Das linke Bild zeigt die Einstellung des Lassos, das rechte Bild das Ergebnis nach dem Loslassen des Mauskopfes.





2.2.5 Hilfslinien einstellen

Unabhängig von den Messkurven lassen sich Hilfslinien im Trend-Fenster einblenden, die es in eine einstellbare Anzahl gleich breiter Bereiche teilen und so einen groben Anhaltspunkt beim Ablesen von Messkurven ermöglichen. Anzahl, Startpunkt und Aussehen der Hilfslinien sind einstellbar.

Weitere Hilfslinien sind den Skalen und Kurven zugeordnet. Ihre Parametrierung wird in Kapitel 3.2.2 und Kapitel 4.3 beschrieben.

2.2.6 Trend-Fenster in Streifen einteilen

Das Trend-Fenster kann in mehrere übereinander liegende Streifen - bzw. bei der Kurvenschreiberdarstellung nebeneinander liegende Streifen - eingeteilt werden. Dazu werden die Werteskalen übereinander gestapelt und die zugeordneten Kurven in den entsprechenden Streifen dargestellt. Wenn eine Kurve einen Wertebereich umfasst, der nicht durch die Skala dargestellt wird, schneidet der Rand des Streifens die Kurve ab.

Es ist auch eine Kombination zwischen Kurven innerhalb von Streifen und Kurven, die über das gesamte Trend-Fenster laufen, möglich. Das Bild in Kapitel 2.2.3 zeigt eine mögliche Kombination.



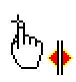
2.2.7 Tooltip

Für alle Skalen und die Kurvenknöpfe kann ein Tooltip eingeschaltet werden.

Der Tooltip zeigt beim Überstreichen einer Skala den Wert an der aktuellen Position des Mauszeigers an. Dadurch können auch Skalenwerte zwischen den Beschriftungen an den Hauptteilungen abgelesen werden. Steht der Mauszeiger auf dem Übersichtsbalken einer Skala, werden minimaler und maximaler Wert der Skala beim Aufschalten und die Grenzen der gesamten Skala angezeigt.

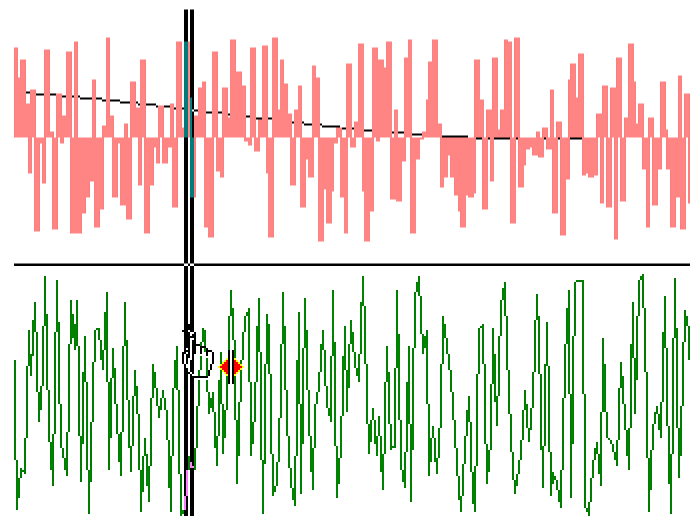
Der Name der jeweiligen Kurve, die mit einem Kurvenknopf bedient wird, sowie der Name der zugeordneten y-Skala erscheinen in einem Tooltip, wenn der Mauszeiger auf dem Kurvenknopf steht.

2.2.8 Lineale und Kurvenknöpfe benutzen



Es gibt zwei interaktive Lineale, die zum Ablesen von einzelnen Kurvenwerten verwendet werden. Das senkrechte x-Lineal liefert immer den Zeitwert an der aktuellen Position, sowie die passenden Messwerte aller eingeschalteten Kurven. Das x-Lineal wirkt für alle Streifen im Diagramm.

Das waagerechte y-Lineal liefert der Anwendung die Skalenwerte aller eingeschalteten Skalen an der aktuellen Position. Die Lineale liegen bei der ersten Benutzung am Rand des Trend-Fensters. Der Cursor zeigt die Lineale an, wenn sie eingeschaltet sind.



Zusätzlich zu den Linealen des Trend-Fensters gibt es für jede Kurve je ein eigenes x- und y-Lineal. Ein Kurvenlineal kann nur über die dafür vorgesehene Schnittstelle eingeblendet werden. Die Bedienung mit dem Mauszeiger erfolgt wie bei den Trend-Fenster Linealen.

Die Kurvenknöpfe werden verwendet, um einzelne Kurven auszublenden. Sie können entweder an den Skalen positioniert werden, denen die entsprechenden Kurven zugeordnet sind oder für alle aktiven Kurven am oberen Rand des ganzen Diagramms zusammengefasst werden.





2.3 Interaktive Benutzung

TrendViewer liefert durch seine dynamisch veränderbaren Komponenten eine hohe Usability für den Benutzer. Die Komponenten können direkt vom Benutzer für die jeweils optimale Ablesbarkeit verändert werden, so dass unterschiedliche Aussagen aus denselben Kurven gewonnen werden können.

Eine grundlegende Eigenschaft des TrendViewer ist die Zoom-Funktion, mit der nahtlos aus Messreihen über lange Zeiträume - z.B. 5 Jahre - detaillierte Kurvenverläufe in sehr kurzen Zeiträumen - z.B. Minuten - herausgeholt werden können. Voraussetzung dazu ist, dass die gesamte Messreihe vorher in den TrendViewer geladen wurde.

Kurven und Skalen korrespondieren auch während des Zoom-Vorgangs miteinander. Die Zoom-Funktion kann sowohl mit jeder Skala als auch mit dem gesamten Trend-Fenster ausgeführt werden. Wenn nur eine Skala verändert wird, sind alle dieser Skala zugeordneten Kurven betroffen. Wenn das Trend-Fenster verändert wird, so sind alle Kurven und Skalen von der Zoom-Funktion betroffen.

Für die Analyse und Dokumentation von Messkurven sind weitere interaktive Eigenschaften des TrendViewer äußerst hilfreich:

1) die dynamische Anordnung von Skalen. Damit können Diagramme in Sekunden neu sortiert und Abhängigkeiten von Messkurven unter verschiedenen Aspekten direkt angesehen werden (Professional License erforderlich)
2) den Vergleich von Messkurvenabschnitten über die Zeit
3) den Vergleich von Messkurven über den Verlauf, um z.B. Soll- und Istkurve zu vergleichen (Professional License erforderlich).
4) das Verändern von Messkurven (Professional License erforderlich)

Alle Funktionen können sowohl über Parameter aus der Anwendung heraus oder interaktiv benutzt werden. Funktion 1 wird in Kapitel 2.3.2, Funktion 2 wird in Kapitel 3., Funktion 3 und 4 werden in Kapitel 4.6 beschrieben.

2.3.1 Einstellen eines Ausschnitts der Messkurve

Bei der Einstellung eines Ausschnitts mit Hilfe einer Werte- oder Zeitskala werden keine zusätzlichen Steuerelemente wie Knöpfe, Rollbalken oder Schieber benötigt, sondern die Skalen werden direkt angefasst und mit den Mausknöpfen in ihrem Bereich eingestellt. Jede Veränderung einer Skaleneinstellung wirkt sich auf die Messkurven im Trend-Fenster aus, so dass die Darstellungen von Skalen und Messkurven stets übereinstimmen.

In Kapitel 3.4 wird die Bedienung der Skalen erläutert.

Zusätzlich kann auch der Inhalt des Trend-Fensters mit der Maus verkleinert, vergrößert oder verschoben werden. Die Skalen werden automatisch angepasst, so dass wiederum die Darstellungen von Skalen und Messkurven konsistent sind.

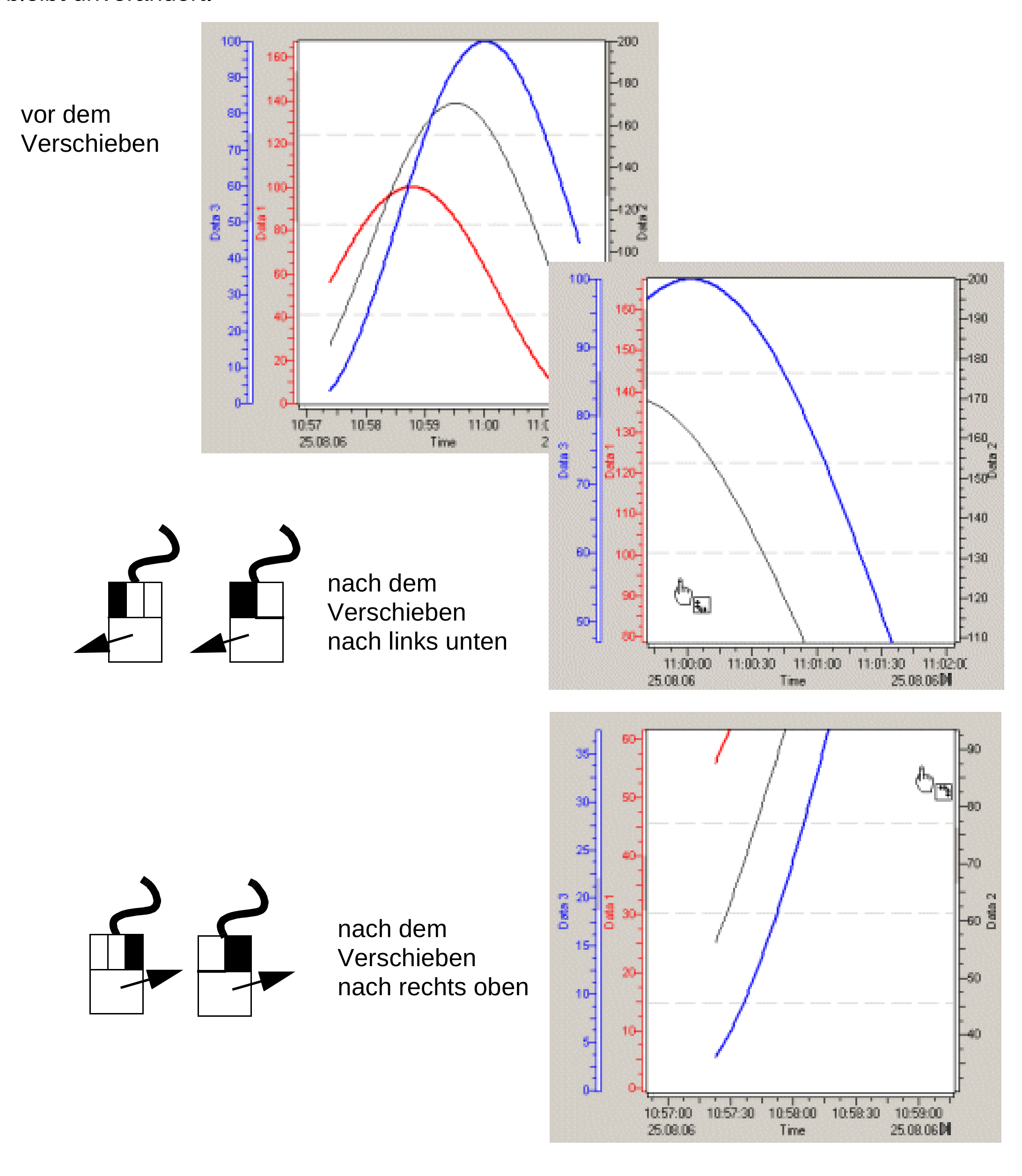
Diese Funktion ist abschaltbar, damit der Benutzer sie nur dann verwendet, wenn sie sinnvoll ist. Versehentliches Bewegen des Trendfeldes verstellt sonst sofort den angewählten Ausschnitt aus der Messdatenmenge.

Das Echo des Cursors zeigt an, ob der dargestellte Ausschnitt aus der gesamten Messdatenmenge verkleinert, vergrößert oder verschoben wird. Alle möglichen Bedienungen werden in den folgenden Abbildungen erläutert.

Verkleinern des Ausschnitts

Das Verkleinern eines Ausschnitts bewirkt die vergrößerte Darstellung von Details aus den Messkurven. Der Ausschnitt kann solange verkleinert werden, bis die geringste Auflösung einer Skala erreicht ist.

Wenn die Maus nach links bewegt wird und gleichzeitig die linke Maustaste gedrückt wird, verschiebt sich der linke untere Punkt des Trendfeldes. Der rechte obere Punkt des Trendfeldes bleibt unverändert.

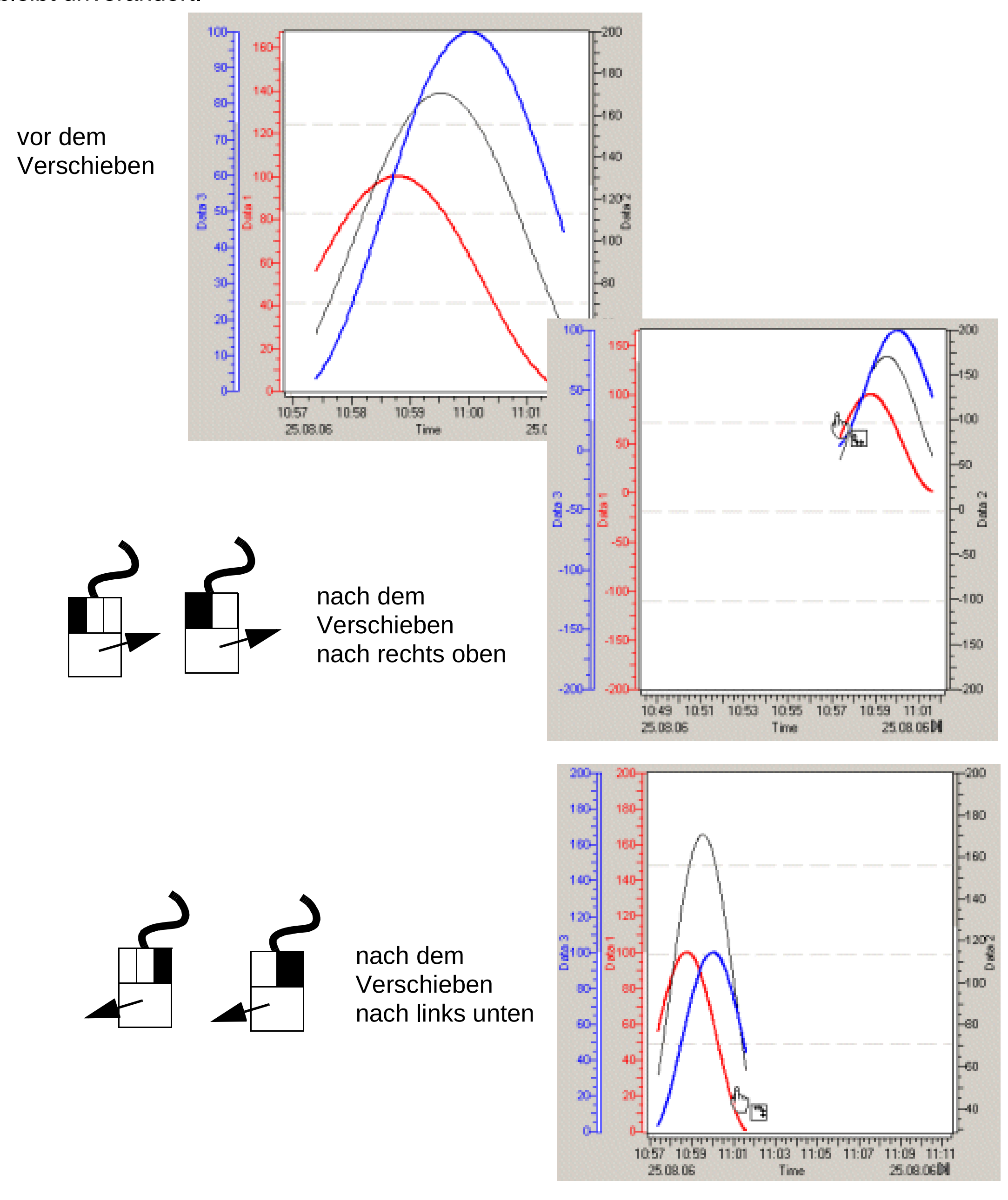


Wenn die Maus nach rechts bewegt wird und gleichzeitig die rechte Maustaste gedrückt wird, verschiebt sich rechte obere Punkt des Trendfeldes. Der linke untere Punkt des Trendfeldes bleibt unverändert.

Vergrößern des Ausschnitts:

Das Vergrößern eines Ausschnitts bewirkt die verkleinerte Darstellung von Details aus den Messkurven. Der Ausschnitt kann solange vergrößert werden, bis die untere bzw. die obere Grenze einer Skala erreicht wird.

Wenn die Maus nach rechts bewegt wird und gleichzeitig die linke Maustaste gedrückt wird, verschiebt sich der linke untere Punkt des Trendfeldes. Der rechte obere Punkt des Trendfeldes bleibt unverändert.

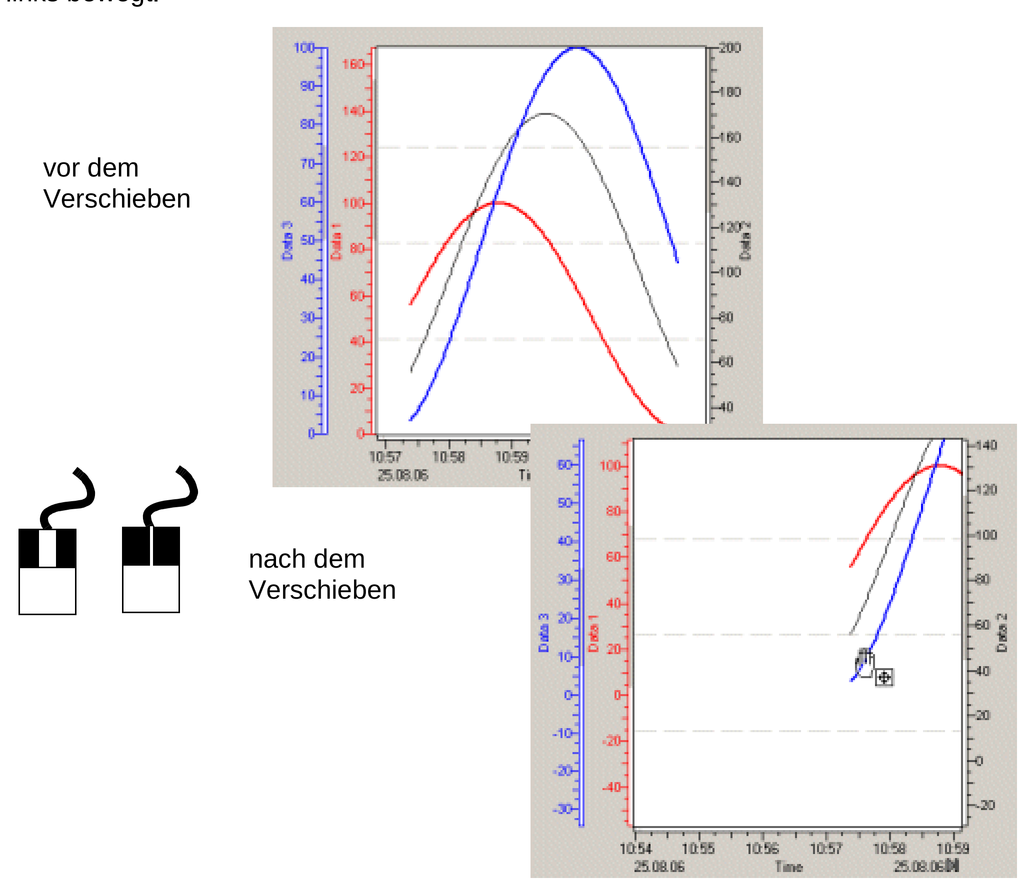


Wenn die Maus nach links bewegt wird und gleichzeitig die rechte Maustaste gedrückt wird, verschiebt sich der rechte obere Punkt des Trendfeldes. Der linke untere Punkt des Trendfeldes bleibt unverändert.

Verschieben des Ausschnitts

Das Verschieben des Ausschnitts bewirkt die Darstellung eines anderen Teiles aller Messkurven, ohne dass die Größe der Details verändert wird.

Die linke und die rechte Maustaste werden gleichzeitig gedrückt und die Maus nach rechts oder nach links bewegt.



2.3.2 Skalen positionieren (Professional License erforderlich)

Die Trend-Komponente bietet zwei x-Skalen und bis zu 512 y-Skalen an. Die x-Skalen sind auf den gegenüberliegenden Seiten des Trend-Fensters angeordnet und können als Zeitskala, lineare oder logarithmische Wertskala oder Grad-Skala konfiguriert werden.

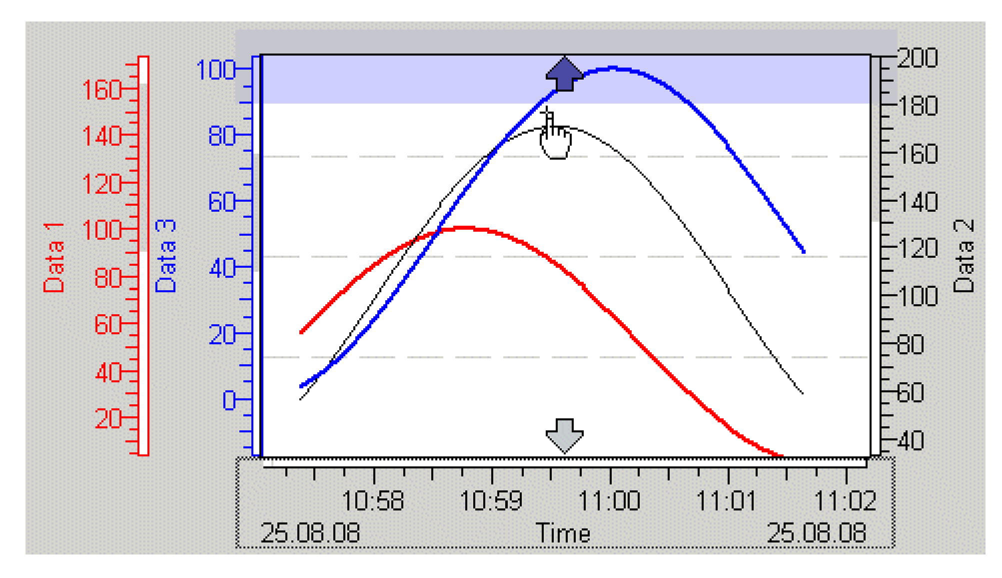
Die y-Skalen sind lineare oder logarithmische Wertskalen oder Grad-Skalen. Sie können entweder spaltenweise nebeneinander liegen oder auch innerhalb einer Spalte übereinander angeordnet sein. Dabei sind beliebige Kombinationen der Positionen möglich.

Um eine x/y-Darstellung zu erreichen, kann eine der y-Skalen auch parallel zu einer x-Skala angeordnet werden. Damit lassen sich z.B. Kurven einer Messstelle in Abhängigkeit zu Kurven anderer Messstellen auftragen (Beispiel: Druck/Temperatur Beziehung).

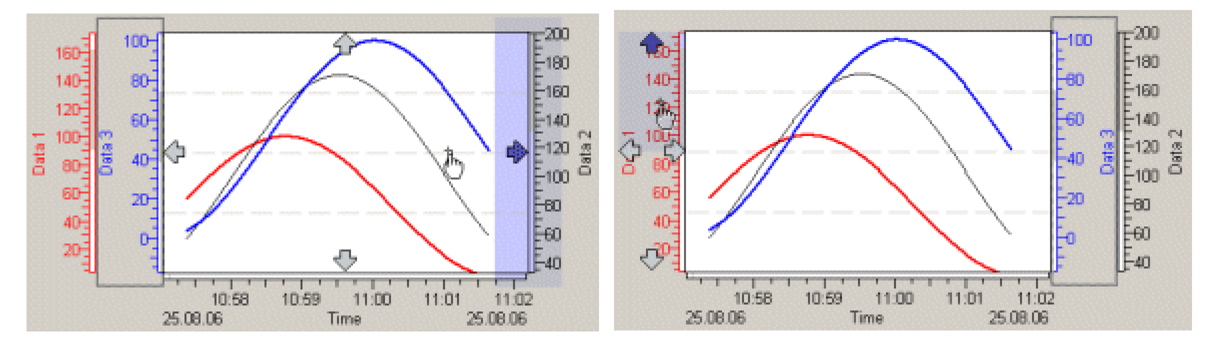
Die Skalen werden an ihrem Knopf oder als Objekt angefasst und an die gewünschte Position bewegt. Dabei gibt TrendViewer durch Echos Hilfestellung.



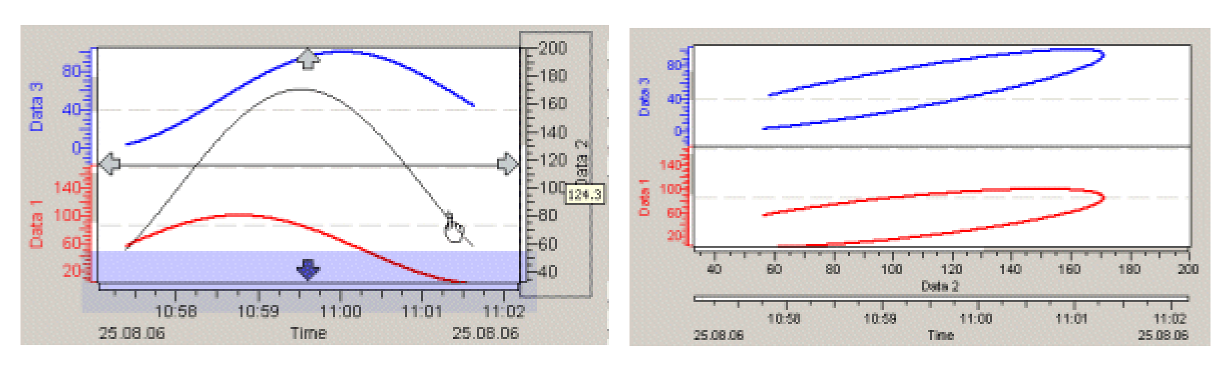
Soll die x-Skala an der gegenüberliegende Seite des Trend-Fensters positioniert werden, zeigt das Echo die beiden möglichen Positionen an:



Soll eine Werteskala an eine andere Position gebracht (Bild unten links) oder über eine andere gestapelt (Bild unten rechts) werden, zeigt das Echo alle möglichen Positionen an, die gerade mit der Maus zu erreichen sind. Der dunkelblaue Pfeil gibt die Position an, die die Skala einnimmt, wenn der Mausknopf losgelassen wird.



Für ein Phasendiagramm kann eine Werteskala auch parallel zur Zeitskala angeordnet werden. Das rechte Bild zeigt ein mögliches Ergebnis nach der Positionierung der Werteskala **Data 2.** Wenn die Kurven gegen **Data 1** statt gegen die Kurve **Data 2** aufgetragen werden sollen, muss nur die Skala **Data 1** an die Position der Skala **Data 2** bewegt werden. **Data 1** und **Data 2** werden damit gegeneinander ausgetauscht.

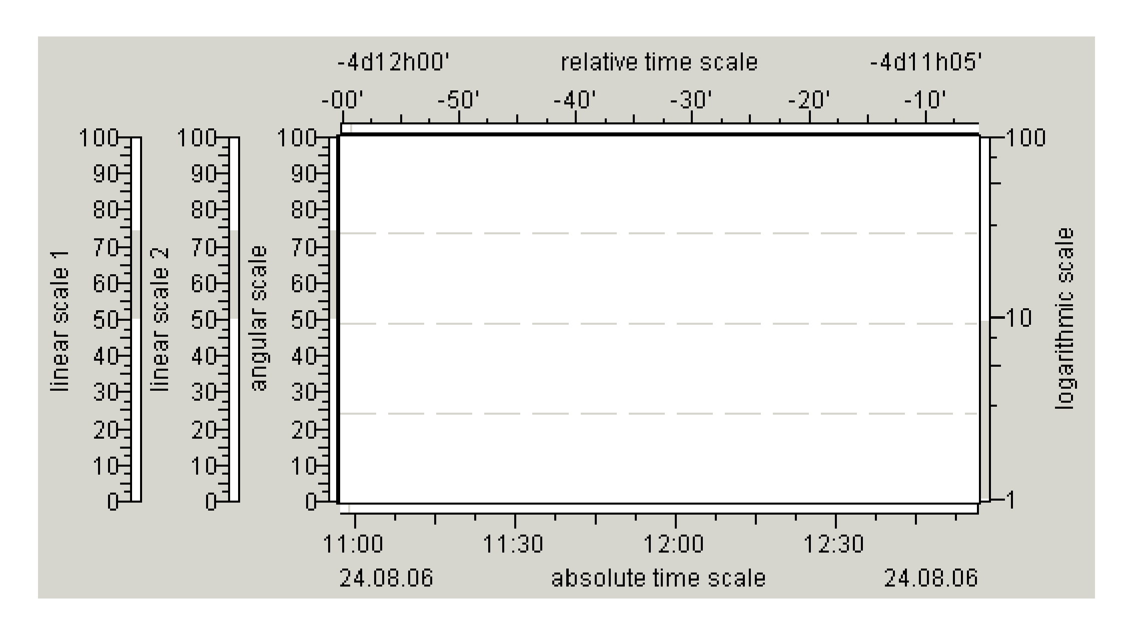




3. Skala

Der wichtigste Bestandteil eines Kurvendiagramms für das exakte Ablesen von Messwerten ist die Skala, die an alle Erfordernisse angepasst werden kann. Das gilt vor allem dann, wenn mit den einzigartigen Zoom-Eigenschaften des TrendViewer ein sehr breites Spektrum von Auflösungen - z.B. für Zeitskalen von Jahren bis Millisekunden - während des Zoomens nahtlos dargestellt werden soll. TrendViewer bietet 5 Skalentypen an, die individuell um das Trend-Fenster herum positioniert werden können.

Die folgende Abbildung zeigt ein Anordnungsbeispiel mit allen Skalentypen:



Es werden 2 X-Achsen angeboten, damit eine Messkurve z.B. gleichzeitig in zwei verschiedenen Zeitintervallen innerhalb eines Diagramms dargestellt werden kann. Beide Skalen lassen sich unabhängig voneinander parametrieren und interaktiv verändern.

Im Phasendiagram wird parallel zu einer Zeitskala eine Werteskala angezeigt, damit das Verhältnis mehrerer Messwerte gegenüber einem bestimmten Messwert im zeitlichen Verlauf beobachtet werden kann.

Jeder Messkurve kann eine eigene Wertskala zugeordnet werden, mehrere Messkurven können auf eine Wertskala bezogen werden. Jede Kurve wird mit einer der beiden Zeitskalen verbunden.

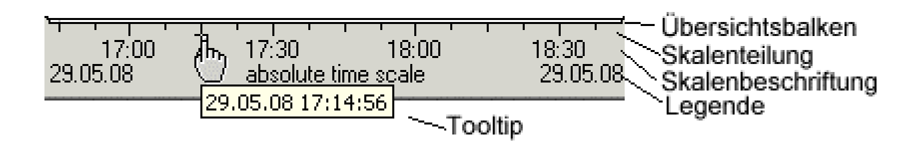
Die Darstellung der Messkurve entspricht immer der aktuellen Skalierung. Wird eine Skala oder die Größe des im Trend-Fenster dargestellten Ausschnitts interaktiv verändert, werden die Messkurven automatisch angepasst.



3.1 Bestandteile einer Skala

Eine Skala besteht aus 2 Tasten und 5 Teilen, die individuell formatiert werden können:

- ein Übersichtsbalken, in dem der im Trend-Fenster angezeigte Ausschnitt im Verhältnis zum gesamten Wertebereich einer Messkurve in Position und Länge markiert wird
- eine Skalenteilung mit Hauptraster, Teilraster und Nebenraster. Die Hauptraster werden mit Werten beschriftet, die Anzahl der Teilraster und Nebenraster kann eingestellt werden.
- die Skalenbeschriftung, die je nach dem Typ der Skala an den Hauptrastern angeordnet wird. Die Schrittweite der Hauptraster wird im Verhältnis zum gesamten dargestellten Intervall in der Datei TVformat.xml definiert. Es wird automatisch dafür gesorgt, das alle Beschriftungen lesbar sind. Wenn der Platz nicht ausreicht, werden Beschriftungen ausgeblendet.
- die Legende, die aus einem Text und bei Zeitskalen aus dem Minimalwert und Maximalwert des Skalenausschnitts besteht. Mit der Anzeige des Minimal- und Maximalwertes - z.B. Datum, Stunde und Minute - kann die Beschriftung an den Hauptrastern z.B. durch Verwendung von Sekunden und Millisekunden so kurz wie möglich gehalten werden.
- der Tooltip, in dem der Wert an der Cursorposition beim Überfahren der Skala angezeigt wird.



Um Platz zu sparen, können der Übersichtsbalken und die Legende ausgeblendet werden. Die Legende eines Diagramms kann dann unabhängig vom *TrendViewer* an einer anderen Stelle durch die Applikation angezeigt werden.

Der Tooltip kann abgeschaltet werden, um bei Systemen mit geringer Prozessorleistung die Performance zu verbessern.

3.1.1 Übersichtsbalken

Der einschaltbare Übersichtsbalken einer Skala zeigt qualitativ das Verhältnis zwischen Wertebereich der zugeordneten Messkurven und dargestelltem Ausschnitt an. Ausserdem wird das Intervall zwischen minimalem und maximalem Wert der Messkurven eingeblendet.

3.1.2 Skalenbeschriftung

Neben der Genauigkeit und dem Textformat einer Skalenbeschriftung, die vom Typ der Skala abhängen, lassen sich vielfältige Parameter wie Schrifttyp, Ausrichtung an der Skala, Schriftrichtung und die Skalenrandbehandlung einstellen. Bei der Formatierung der Skalenbeschriftung muss vor allem ausreichender Platz bereitgestellt werden, damit keine unvollständigen Beschriftungen erscheinen, sowie für eine optimale Ablesbarkeit gesorgt werden.

Bei Konflikten zwischen dem zur Verfügung stehenden Platz und optimaler Ablesbarkeit muss für jeden Fall entschieden werden, welche Forderung den Vorrang hat. Konflikte treten z.B. bei direkt aneinanderliegenden Skalen (sogenannte gestapelte Werteskalen) oder dann auf, wenn die Trenddarstellung in einem Fenster liegt, dessen Größe interaktiv verändert werden kann. *TrendViewer* versucht, bei Veränderung der Fenstergröße immer, eine möglichst gute Darstellung zu gewährleisten.

3.2 Skalentypen

Es wird grundsätzlich zwischen Werte- und Zeitskalen unterschieden. Werteskalen werden in Zeit-Wertedarstellungen (y,t-Diagramme) von analogen und binären Signalen für die Y-Achsen eines Diagramms verwendet. Eine Ausnahme bilden die Ortskurven (x,y-Diagramme) oder Phasendiagramme (x,y,t-Diagramme), in denen Werteskalen auch als X-Achsen verwendet werden.

3.2.1 Werteskala

Eine Werteskala kann eine lineare, logarithmische oder eine Winkel-Teilung annehmen. Die Umschaltung zwischen linearer und logarithmischer Teilung ist auch in der Applikation während der Benutzung der Skala möglich, wenn es die Skalenanfangswerte und -endwerte zulassen. Die Winkel-Teilung kann zyklisch über 360 Grad oder linear dargestellt werden.

Um in jeder Situation eine optimale Ablesbarkeit zu gewährleisten, kann das Format der Skalenbeschriftung für die Hauptraster in der Datei TVformat.xml eingestellt werden. Für lineare und Winkel-Teilung ist jeweils ein Abschnitt enthalten, in dem für jeden Ausschnitt der Messkurven, der mit einer Werteskala dargestellt wird, das Format der Beschriftungen für Hauptraster, Legende und Tooltip festgelegt wird.

Folgendes Beispiel zeigt einen Ausschnitt der Datei TVformat.xml:

```
<DecimalFormat>
<RangeValue>20.2</RangeValue>
<MainGrid>5</MainGrid>
<ScaleFormat>%.lf</ScaleFormat>
<LegendFormat></LegendFormat>
<TooltipFormat>%.1f</TooltipFormat>
</DecimalFormat>

<DecimalFormat>
<RangeValue>10.1</RangeValue>
<MainGrid>2</MainGrid>
<ScaleFormat>%.lf</ScaleFormat>
<LegendFormat></LegendFormat>
<TooltipFormat>%.1f</TooltipFormat>
</DecimalFormat>
```

Mit <RangeValue> wird der mit der Skala beschriftete Ausschnitt aus dem Wertebereich bezeichnet, der im Diagramm zu sehen ist.

<MainGrid> gibt den Wertebereich an, der einem Hauptraster der Skala entspricht. Dieser Wertebereich ist ab dem <RangeValue>, dem <MainGrid> zugeordnet ist bis zum nächst höheren <RangeValue> gültig. Im Beispiel wird das <MainGrid> =2 beginnend mit dem <RangeValue> > 10.1 bis zum <RangeValue> <= 20.2 gesetzt.

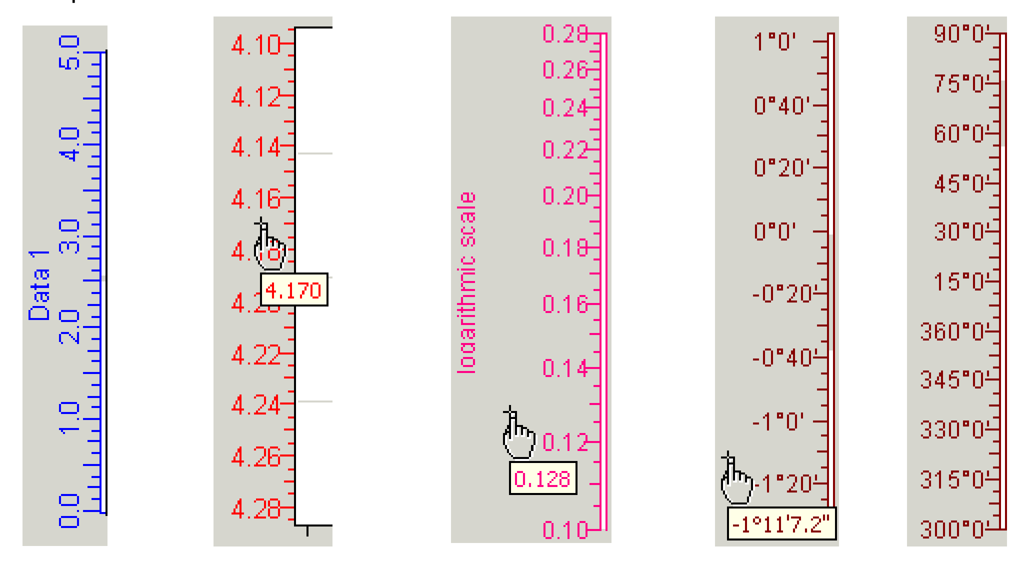
Es wurde immer ein 1% größerer <RangeValue> als der gewünschte maximale Wert eingetragen, damit Abbildungsungenauigkeiten auf dem Monitor insbesondere beim interaktiven Setzen der Skalenanfangs- bzw. -endwerte kompensiert werden.

<ScaleFormat>, <LegendFormat> und <TooltipFormat> geben das Format der Skalenbeschriftung, der Legende und des Tooltips an (C-Syntax der Formatanweisung). Das <TooltipFormat> ist immer eine Stelle genauer als das <ScaleFormat> angegeben, damit beim Überstreichen mit dem Cursor eine genauere Wertangabe dargestellt werden kann.



Es wird eine TVformat.xml Datei mit *TrendViewer* ausgeliefert, die der Standardeinstellung entspricht. Die Datei kann aber nach den Erfordernissen der Applikation geändert werden.

Beispiele für Werteskalen:



3.2.2 Zeitskala

Sowohl Zeitabschnitte (z.B. Meßwerte ab Beginn einer Charge) als auch absolute Zeiten mit Datum und Uhrzeit lassen sich anzeigen.

Auch bei der Zeitskala wird immer eine möglichst große Genauigkeit bei der Beschriftung der Skalenabschnitte angestrebt. Da die Beschriftung der Hauptraster mit Datum (Tag, Monat, Jahr) und Uhrzeit (Stunde, Minute, Sekunde, Millisekunde) sehr lang wird, kann sie auf die Legende erweitert werden. In der Legende stehen minimaler und maximaler Zeitwert des dargestellten Intervalls.

Die Genauigkeit der Beschriftung kann der Größe des jeweils dargestellten Intervalls angepasst werden. Zeigt die Skala z.B. nur wenige Sekunden an, kann die Skalenteilung auf Millisekunden eingestellt werden. Werden Jahre angezeigt, kann die Skalenteilung z.B.auf 90 Tage eingestellt werden.

Das länderspezifische Datumsformat wird in der TVformat.xml Datei festgelegt. Reihenfolge der Datumsbestandteile und Trennzeichen können für Tooltip und Skalenbeschriftung sogar getrennt festgelegt werden.

Die Umstellung auf die Sommerzeit wird berücksichtigt.



Skalen

Beispiele für die Zeitskala

Minutendarstellung

17:34 17:36 17:38 17:40 17:42 17:44 17:46 17:48 17:50
29.05.08 absolute time scale 29.05.08

Sekundendarstellung

14.0 16.0 18.0 20.0 22.0 24.0 26.0 28.0 30.0 32.0
29.05.08 17:50:13 absolute time scale 29.05.08 17:50:32
29.05.08 17:50:18.381

Tooltip mit englischem Datumsformat

21.04. 25.04. 29.04. 03.05. 07.05.
19.04.08 absolute time scale 07.05.08
04/22/08 23:54

ein halbes Jahr mit 30 Tage Raster

01.04. 01.05. 31.05. 30.06. 30.07. 29.08.
22.03.2008 absolute time scale 20.09.2008
15.06.2008

mehr als 100 Jahre

28.10.1923 21.11.1964 16.12.2005 10.01.2047 04.02.2088
absolute time scale

relative Zeitskala

-00h10' 00h00' 00h10' 00h20' 00h30'
30.06.08 relative time scale 30.06.08
00h25'



3.3 Skaleneigenschaften

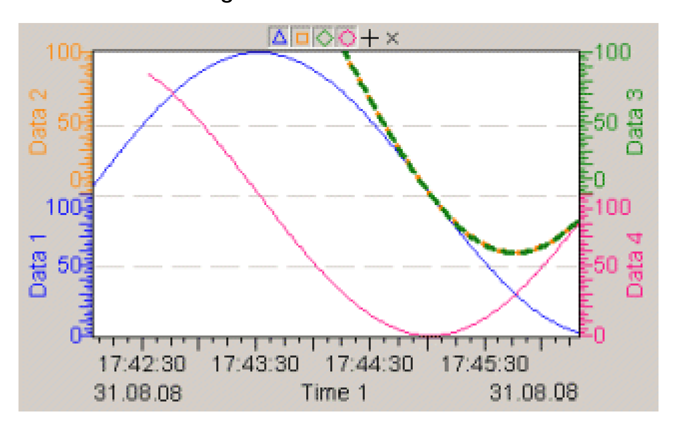
Außer den grafischen Attributen wie Farbe und Beschriftung können Skalen auch noch zusätzliche Komponenten wie den 'Follow-Me' Knopf für das Fortschreiben einer online aufgezeichneten Messkurve und Skalenknöpfe zum Ein- und Ausblenden der zugeordneten Messkurven und zur Information über die Kurvennamen enthalten.

Die Richtung der Beschriftung ist für alle Skalentypen wählbar, so dass z.B. online aufgezeichnete Messkurven von links nach rechts oder von rechts nach links geschrieben werden.

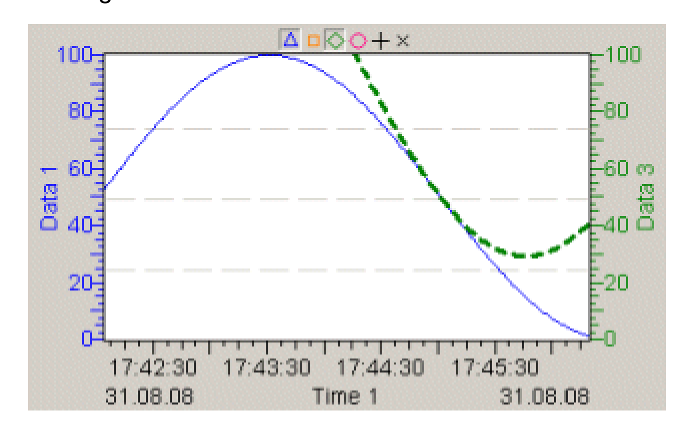
3.3.1 Skalenanzeige und -positionen

Die Position kann sowohl interaktiv, wie in Kapitel 2.3.2 beschrieben, verändert als auch durch Parameter bestimmt werden. Die x-Skalen sind oben und unten an einem Diagramm zu positionieren, wobei beide x-Skalen auch übereinander an einer Seite des Diagramms liegen können.

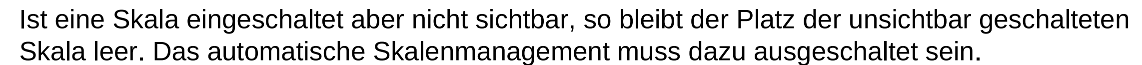
Die y-Skalen werden links oder rechts neben dem Diagramm dargestellt, wobei mehrere y-Skalen nebeneinander und übereinander gesetzt werden können.



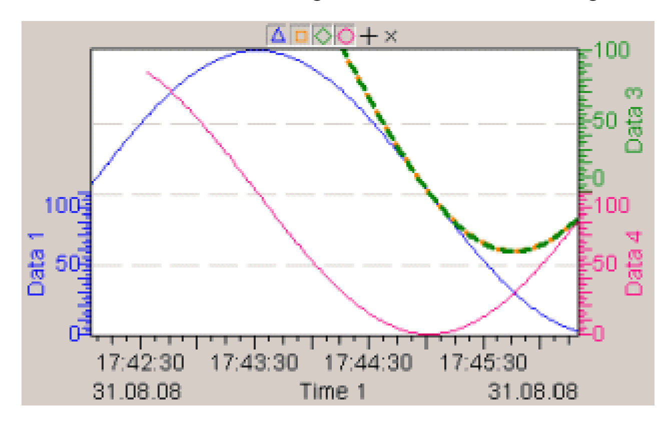
Ist das automatische Skalen-Management eingeschaltet, werden immer nur Skalen dargestellt, denen Kurven zugeordnet sind. Auf diese Weise erhält man immer ein Diagramm mit minimaler Information, wenn z.B. über die Kurvenknöpfe Kurven ein- und ausgeschaltet werden. Im Bild sind die Kurven 2 und 4 ausgeschaltet



Die Darstellung einer Skala wird durch die Attribute 'eingeschaltet' und 'sichtbar' zusätzlich gesteuert. Wenn bei übereinander gesetzten Skalen eine der Skalen ausgeschaltet wird, so expandiert die angrenzende Skala in den Bereich der ausgeschalteten Skala.



Ist eine Skala eingeschaltet aber nicht sichtbar, so bleibt der Platz der unsichtbar geschalteten Skala leer. Das automatische Skalenmanagement muss dazu ausgeschaltet sein.



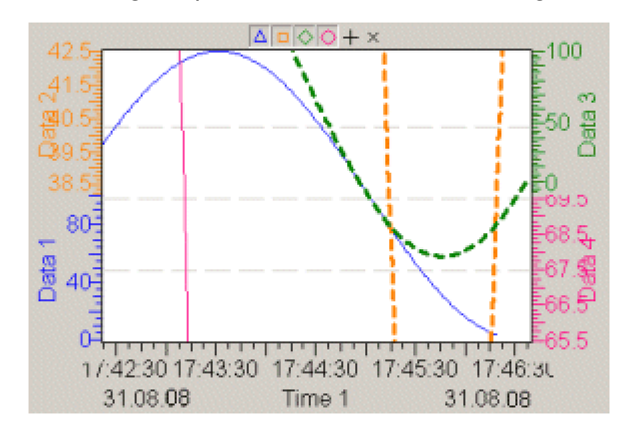
3.3.2 Grafische Attribute

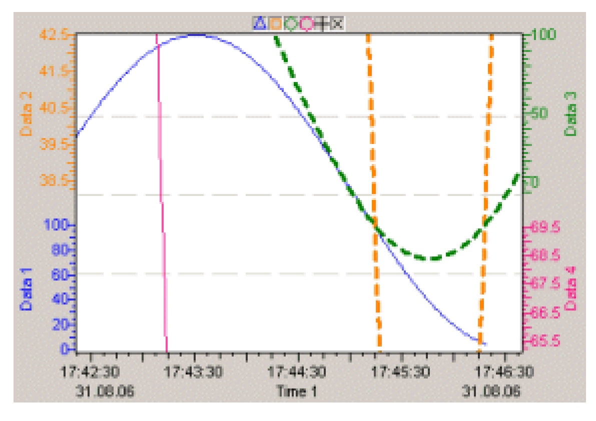
Über die Farbe läßt sich die Zuordnung zwischen Skala und Kurve sichtbar machen. Farben sind beliebig einstellbar, jedoch sollte auf ausreichenden Kontrast zum Bildhintergrund geachtet werden, um die Lesbarkeit zu gewährleisten.

An die Beschriftung einer Skala sind wegen der Informationsmenge, die angezeigt werden muss, hohe Anforderungen gestellt. Vor allem, wenn die Beschriftung während des Zoom-Vorgangs jederzeit ablesbar sein muss und nur beschränkter Platz zur Verfügung steht, muss darauf geachtet werden, dass vollständige Informationseinheiten zu sehen sind. *TrendViewer* versucht dies soweit als möglich automatisch zu gewährleisten.

Der für die Beschriftung verfügbare Platz wird durch die Größe des Diagramms, den Platz, den das Trend-Fenster benötigt, den Font, den Abstand zwischen den Hauptrastern einer Skala und die Schriftrichtung bestimmt. Wenn der zur Verfügung stehende Platz zunächst nicht ausreicht, kann durch Verkleinern des Fonts, Vergrößern des Abstands zwischen Skala und Legende (bei waagerechter Beschriftung) und Margins am Skalenrand mehr Platz geschaffen.

Bei direkt aneinander grenzenden Skalen wie z.B. bei übereinander gesetzten Skalen, kann durch Einschieben unsichtbarer Skalen und durch Vergabe von Prioritäten die Darstellung der Beschriftungen optimiert werden. Das Bild zeigt ein Diagramm vor und nach der Platzoptimierung.





Wenn der Platz zwischen den Hauptrastern oder am Rand der Skala nicht ausreicht, werden Beschriftungen automatisch weggelassen. Das kann durch eine entsprechende Formatierung der Beschriftung in der Datei TVformat.xml angepasst werden.



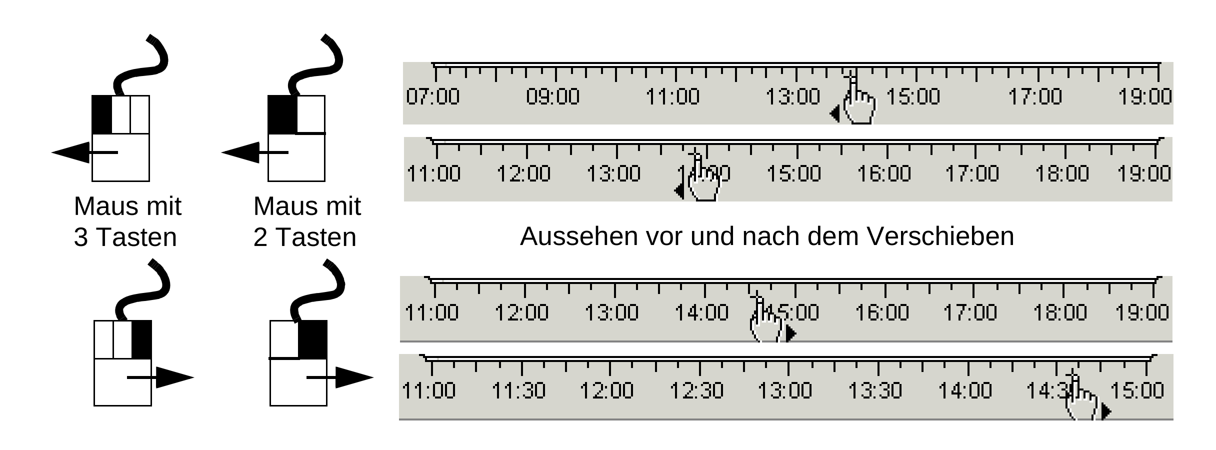
3.4 Skalenausschnitt interaktiv verändern

Mit einer Maus oder einem vergleichbaren Positioniergerät kann der Skalenausschnitt direkt verändert werden.

Das Echo des Cursors zeigt an, ob der dargestellte Ausschnitt aus dem gesamten Wertebereich der Messkurven verkleinert, vergrößert oder verschoben wird. *TrendViewer* sorgt immer dafür, dass alle einer Skala zugeordneten Kurven im Trend-Fenster proportional mit der Skala verändert werden.

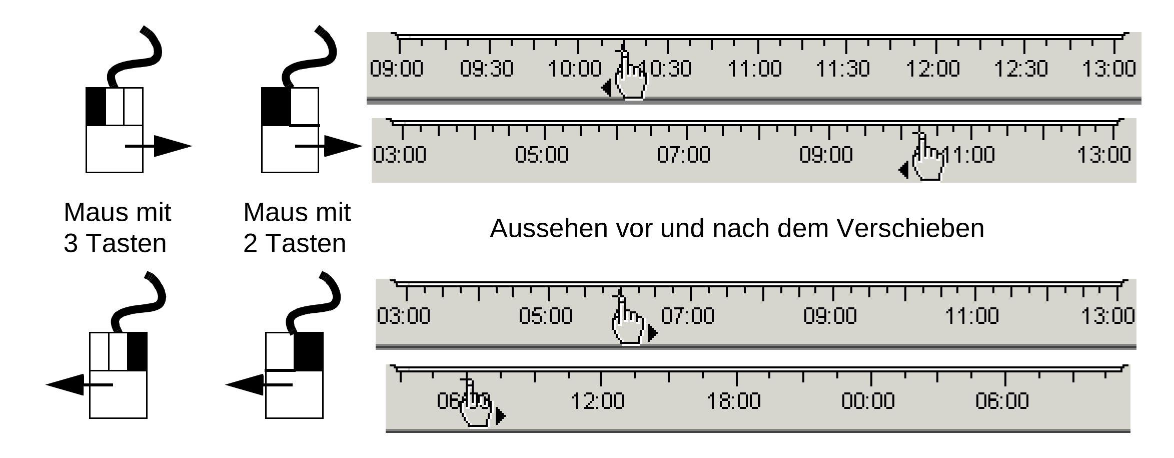
Mit der linken Maustaste wird der linke Rand einer X-Skala bzw. der untere Rand einer Y-Skala verändert. Mit der rechten Maustaste wird der rechte Rand einer X-Skala bzw. der obere Rand einer Y-Skala verändert. Der jeweils gegenüber liegende Rand bleibt unverändert.

Verkleinern eines Ausschnitts (Zoom In)



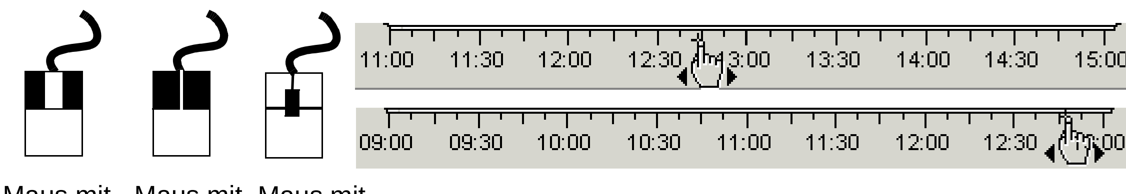
Das Verkleinern eines Ausschnitts bewirkt die vergrößerte Darstellung von Details aus dem gesamten Wertebereich der Messkurven. Der Ausschnitt läßt sich solange verkleinern, bis die eingestellte minimale Auflösung der Skalendarstellung erreicht ist.

Vergrößern eines Ausschnitts (Zoom Out)



Das Vergrößern eines Ausschnitts bewirkt die verkleinerte Darstellung von Details aus dem gesamten Wertebereich der Messkurven. Der Ausschnitt kann solange vergrößert werden, bis die maximale obere Grenze bzw. die minimale untere Grenze der Skalendarstellung erreicht ist.

Verschieben eines Ausschnitts (Panning):



Maus mit 3 Tasten | Maus mit 2 Tasten | Maus mit Rad

Aussehen vor und nach dem Verschieben

Das Verschieben des Ausschnitts bewirkt die Darstellung eines anderen Teiles aus dem gesamten Wertebereich der Messkurven, ohne daß die Größe der Details verändert wird. Der Ausschnitt kann solange verschoben werden, bis die Grenzen der Skalendarstellung erreicht werden.

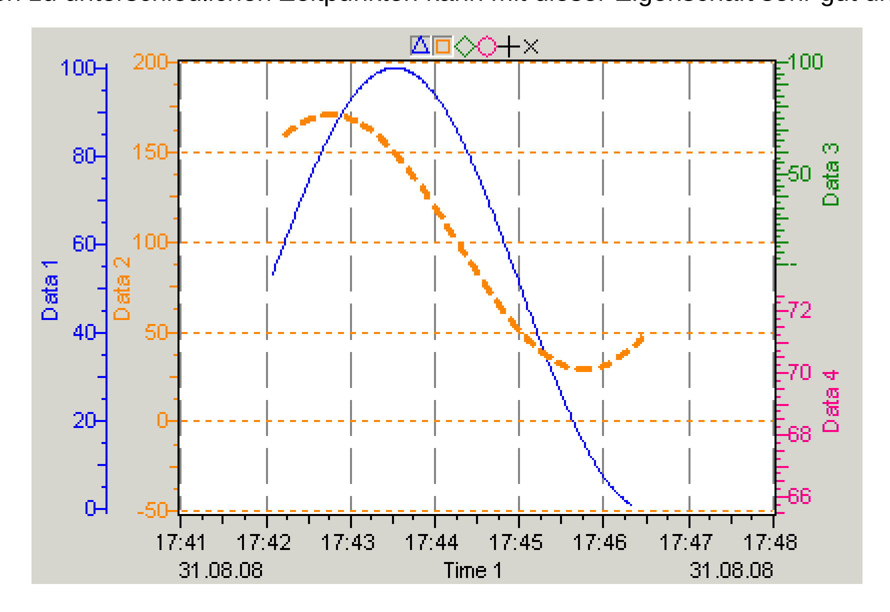
Die linke und die rechte Maustaste werden gleichzeitig gedrückt und die Maus nach rechts oder nach links bewegt. Durch eine entsprechende Konfiguration der Skala (siehe **MoveOnly**) kann das Verschieben des Ausschnitts auch mit einer beliebigen Maustaste erreicht werden.

Um ein unbeabsichtigtes Verschieben der Skalen zu verhindern, kann das Bedienkonzept umgeschaltet werden. Für die Veränderung des Ausschnitts muss dann zusätzlich zur Bewegung der Maus die Umschalt-, Strg- oder Alt-Taste gedrückt gehalten werden.



3.5 Hilfslinien einblenden

Um die Ablesbarkeit der Kurven zu verbessern, können Hilfslinien für jede Skala eingeblendet werden. Besonders hilfreich ist diese Eigenschaft, wenn die beiden gegenüber liegenden x-Skalen mit Hilfe der Zoom-Funktion zur Deckung gebracht werden sollen. Der Vergleich von Kurvenverläufen zu unterschiedlichen Zeitpunkten kann mit dieser Eigenschaft sehr gut unterstützt werden.



Der Abstand zwischen den Hilfslinien und die Anzahl der Hilfslinien kann eingestellt werden. Bei allen Skalen ist auch eine beliebige Anzahl (z.B. jede Stunde) von Hilfslinien einstellbar.



4. Kurve

TrendViewer bietet 2 Kurventypen mit unterschiedlichen Eigenschaften an: binäre und analoge Kurven. Den Kurven sind Attribute zugeordnet, die die Darstellung z.B. im Linienstil, der Farbe, Strichstärke und Interpolation zwischen den Stützstellen beeinflussen. Ausserdem kann die Fläche zwischen einer Kurve und einer Grenzlinie oder zwischen zwei Kurven geflutet werden. Auch die Darstellung der Flutfläche ist einstellbar. Kurven werden Skalen zugeordnet, um so Gruppen zu bilden, die das Verhältnis zwischen Messkurven zeigen.

TrendViewer puffert alle Kurvenwerte, die zur Darstellung übergeben werden. Wenn z.B. in einer Maschinensteuerung Messdaten fortlaufend mitgeschrieben und dargestellt werden, wächst der Puffer ständig an und beeinträchtigt die Leistung der Maschinensteuerung nach längerer Laufzeit.

Die Länge des Kurvenpuffers kann daher für jede Kurve eingestellt werden. Der Puffer wird dann zyklisch beschrieben.

Für die Anzeige von Phasendiagrammen ist es erforderlich, für alle Kurven dieselben x-Werte zu verwenden, damit die y-Werte der Messkurven zueinander in Beziehung gesetzt werden können. Es wird davon ausgegangen, dass die Applikation dafür sorgt, dass die Messkurven zur gleichen Rasterung der x-Werte - das sind meistens die Zeitwerte der Kurven - an den *TrendViewer* übergeben werden.



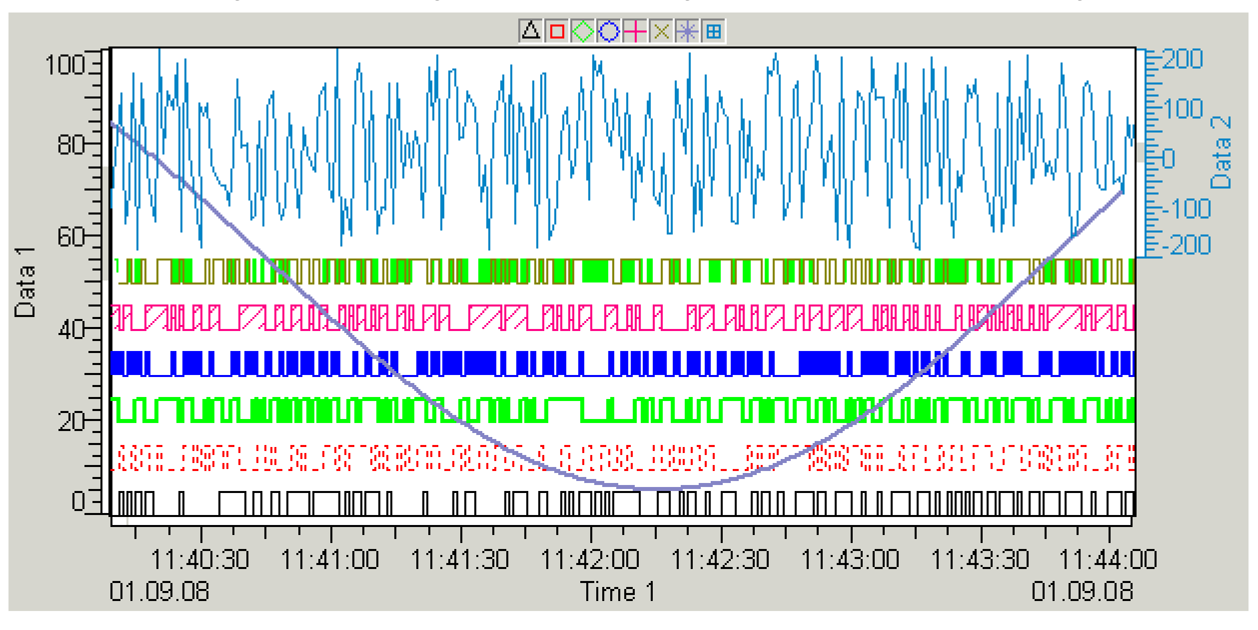
4.1 Kurventypen

Analoge Kurven zeigen Wertverläufe von analogen Messstellen. Jeder Messwert wird durch eine Stützstelle der Kurve dargestellt. Der Wertebereich und die Auflösung eines Messwertes werden durch die Messstelle vorgegeben. Der Verlauf der Kurve wird in Übereinstimmung mit der zugeordneten Skala dargestellt und läßt sich so bewerten.

Stützstellen von binären Kurven können nur die Werte 0 und 1 annehmen. Kurvenwerte < 0 werden auf 0 gesetzt, Werte > 1 werden auf 1 gesetzt.

Binärkurven sind von analogen Kurven unabhängig. Lediglich die Zuordnung zu einer gemeinsamen X-Achse stellt einen Bezug zwischen binären und analogen Kurven her.

In der folgenden Abbildung sind mehrere analoge und binäre Kurven zusammengestellt.



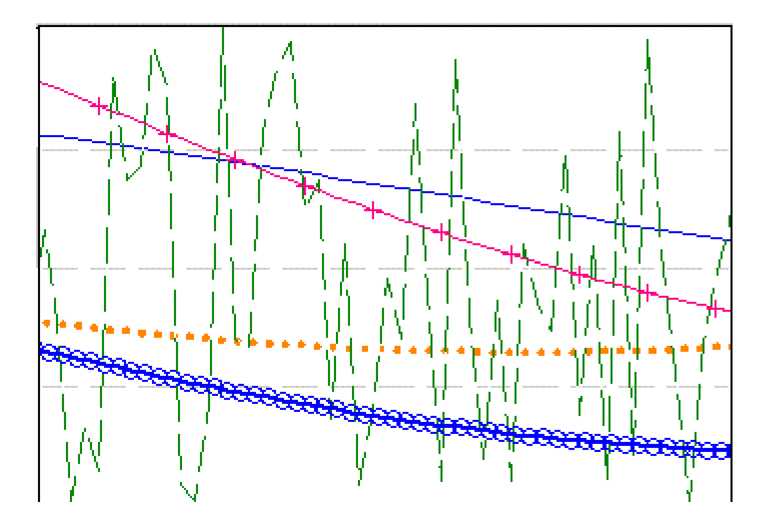


4.2 Darstellung

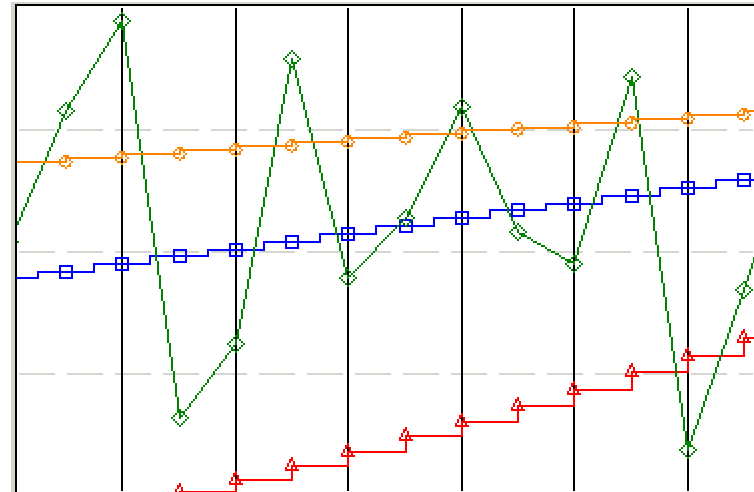
Unterschiedliche grafische Darstellungen von Kurven dienen dazu, sie in einem Diagramm unterscheiden zu können oder sie optisch zusammenzufassen. Wenn Diagramme auf einem Drucker ausgegeben werden, der keine Farbinformation abbilden kann, muss durch unterschiedliche Symbole und Muster für die Lesbarkeit gesorgt werden.

4.2.1 Elementare Eigenschaften

In der folgenden Abbildung sind alle Attribute, die eine Kurve annehmen kann, enthalten. Es kann eingestellt werden, wieviele Stützstellen (jede, jede zweite usw.) einer Kurve ein Symbol erhalten sollen.



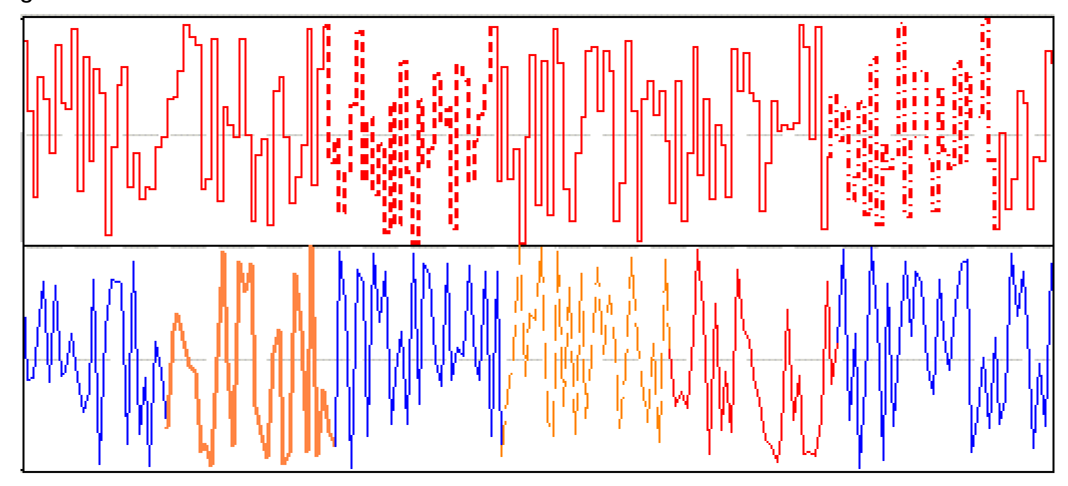
Die Interpolation zwischen 2 Stützstellen kann 4 Formen annehmen, die in der folgenden Abbildung dargestellt werden. Bei Treppenkurven kann die Treppenstufe an der linken, der rechten oder zwischen zwei Stützstellen liegen. Jede Stützstelle wird mit einem Symbol markiert, damit die Unterschiede sichtbar werden. Das Symbol kann vom Entwickler vorgegeben werden.



4.2.2 Status

Jeder Stützstelle einer Kurve ist ein Zustandswert zugeordnet, der durch eine entsprechende grafische Darstellung der Verbindungslinie mit Farbe, Strichstärke oder Linienstil unterschieden wird.

Damit kann innerhalb einer Messkurve z.B. zwischen korrekten Messwerten, Ersatzwerten und gestörten Werten unterschieden werden.

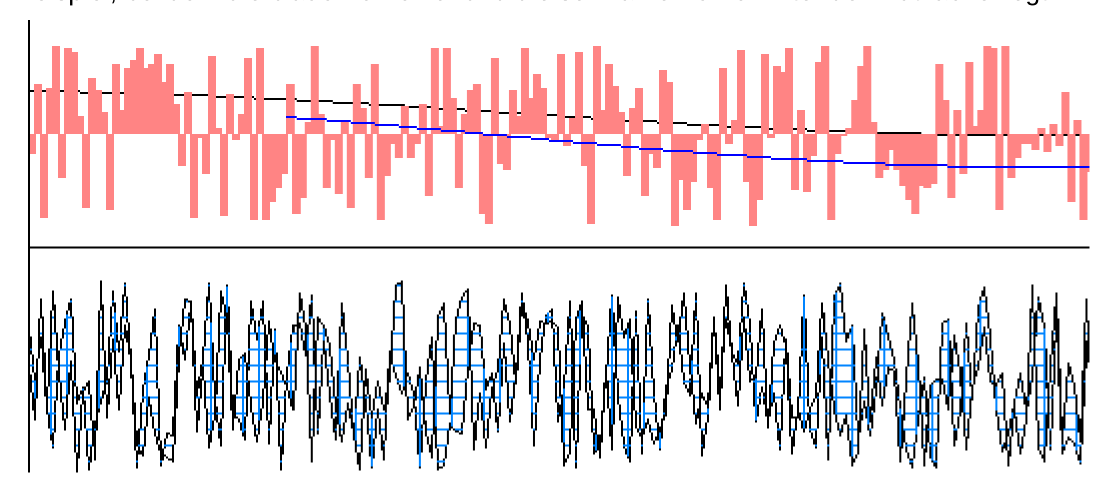


Statuswerte können sowohl analogen als auch binären Kurven zugeordnet werden.

4.2.3 Fluten

Um in komplexen Diagrammen die Abweichung zwischen zwei Messkurven oder dem Abstand einer Messkurve von einem Grenzwert besser sichtbar zu machen, kann die Flutfunktion eingesetzt werden. Die Flutfläche läßt sich mit eigener Farbe und eigenem Muster versehen, damit die Unterscheidung zu den übrigen grafischen Elementen im Diagramm verbessert wird.

Im Zusammenhang mit Flutflächen ist die Vorgabe von Zeichenprioritäten erforderlich, damit Messkurven nicht von Flutflächen verdeckt werden. Die folgende Abbildung zeigt im oberen Teil ein Beispiel, bei dem die blaue Kurve vor und die schwarze Kurve hinter der Flutfläche liegt.



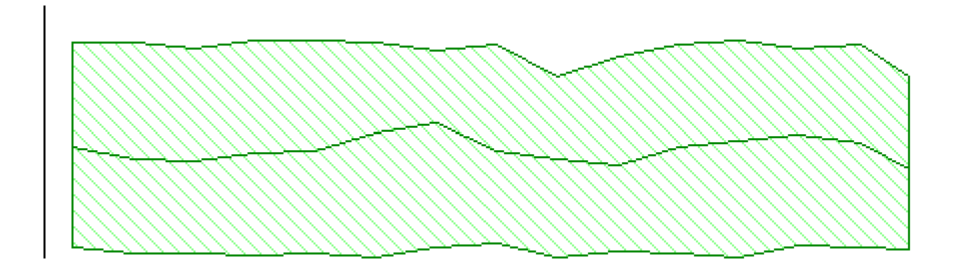
Fluten ist nur jeweils zwischen analogen und zwischen binären Kurven möglich.

Die Kurven können denselben oder unterschiedlichen x- und y- Skalen zugeordnet sein.



4.2.4 Hüllkurve

Bei besonderen Aufgabenstellungen, die keine detaillierte Darstellung des Messkurve erfordern, sondern bei denen die Angabe eines Bandes, in dem sich die Messkurve befindet, ausreicht, kann eine Hüllkurvendarstellung eingesetzt werden. Zusätzlich zur Hüllkurve läßt sich eine Mittelwertlinie einblenden.



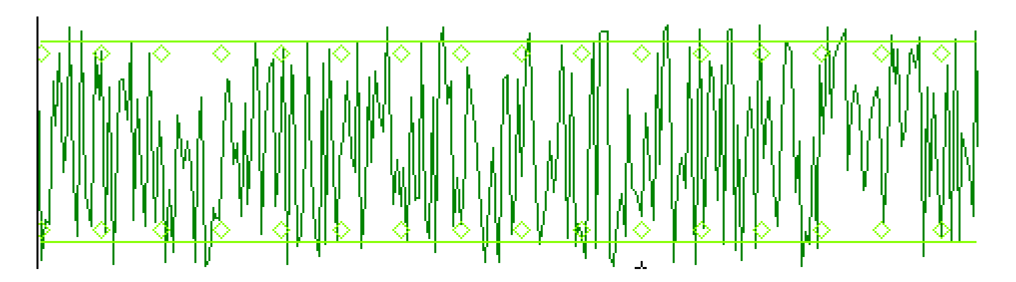
Hüllkurven können nur bei analogen Kurven eingestellt werden.



4.3 Grenzwertlinien

Ein weiteres Hilfmittel zur Beurteilung von Messkurven sind Grenzwertlinien, die zu jeder Messkurve als maximaler und minimaler erwarteter Wert eingeblendet werden können. Grenzwert-über- oder -unterschreitungen sind so leicht zu erkennen.

Um Grenzwertlinien mehrerer Kurven unterscheiden zu können, werden sie mit dem der Kurve zugeordneten Symbol gekennzeichnet. Symbole sind unterhalb der oberen Grenzwertlinie und oberhalb der unteren Grenzwertlinien gezeichnet. Die Linienattribute können individuell eingestellt werden.



Grenzwertlinien lassen sich mit der Maus verschieben.

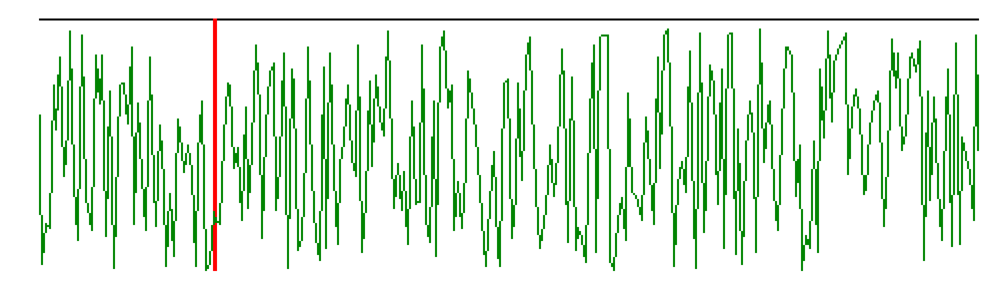
Binäre Kurven haben keine Hilfslinien.



4.4 Lineale

Jeder Kurve ist ein waagerechtes und ein senkrechtes Lineal zugeordnet. Die Lineale werden vom Rand des Trend-Fensters mit der Maus ins Diagramm gezogen. Es ist jeweils nur das waagerechte oder das senkrechte Lineal zu sehen und zu bedienen. Das Lineal des Diagramms (siehe Kapitel 2.2.8) muss ausgeschaltet sein.

Die Linienattribute der Lineale lassen sich individuell einstellen, um mehrere Lineale unterscheiden zu können.



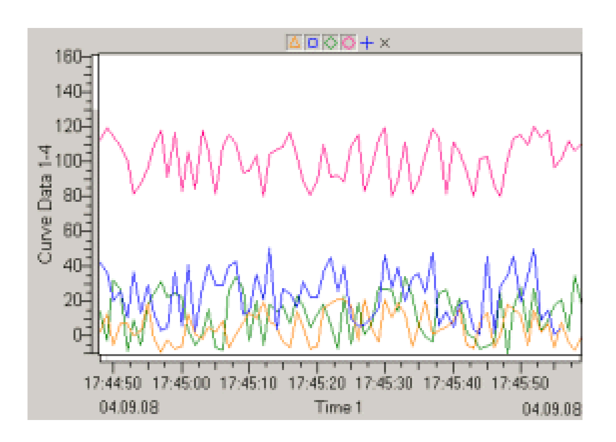
Wird das senkrechte Lineal mit der Maus bewegt, so erhält die Applikation fortlaufend die X-Koordinate und den Wert der Messkurve an der Linealposition. Diese Werte kann die Applikation für den Benutzer darstellen.

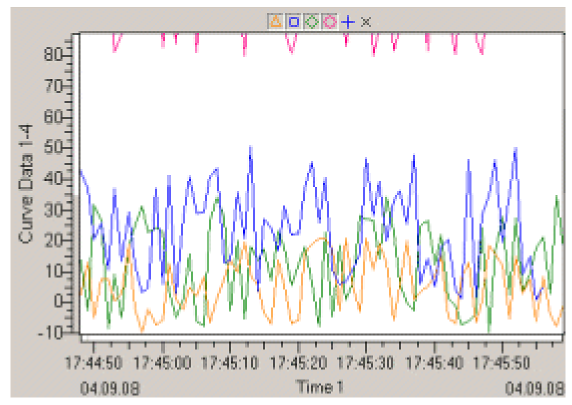
Binäre Kurven haben nur ein senkrechtes Lineal, um damit den binären Wert zu einem bestimmten Zeitpunkt erhalten zu können.

4.5 Skalenzuordnung

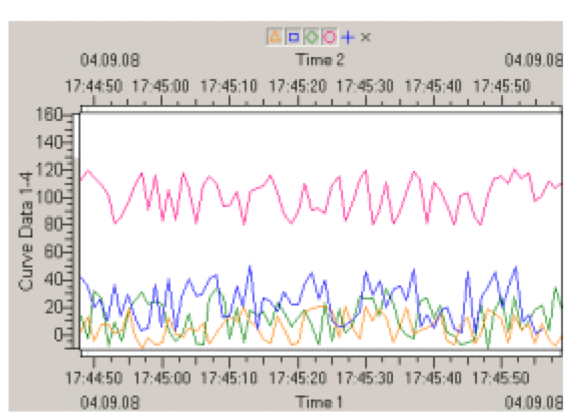
Jede Kurve muss einer der beiden x-Skalen und einer der 512 y-Skalen zugeordnet werden.

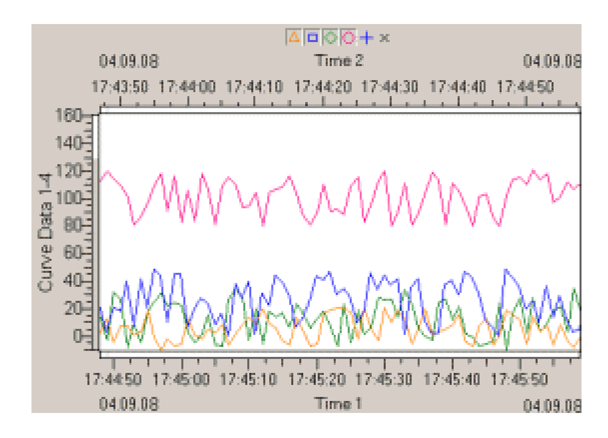
Mehrere Kurven können derselben Skala zugeordnet werden. Dadurch wird die Vergleichbarkeit von Kurvenverläufen für mehrere Messstellen unterstützt. Die Zoom-Funktion wirkt auf alle Kurven, die einer Skala zugeordnet sind, gleichzeitig. In der Abbildung wird das Skalenende von Curve Data 1-4 von 160 auf 80 reduziert.





Im folgenden Bild wurde die blaue Kurve der zweiten x-Skala zugeordnet und um eine Minute verschoben.







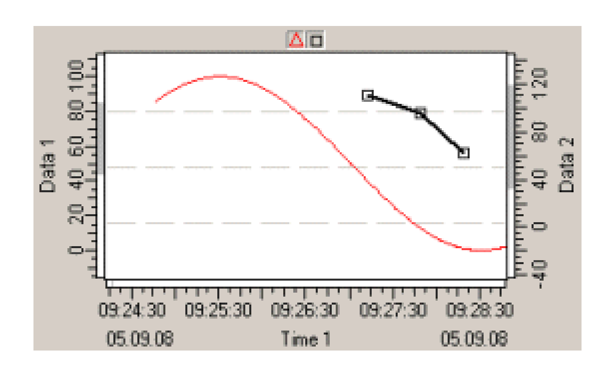
4.6 Kurven bearbeiten (Professional License erforderlich)

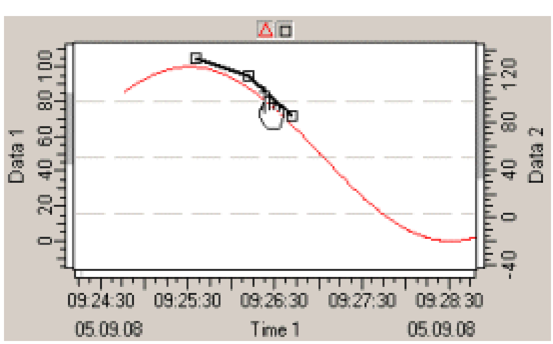
Eine Aufgabenstellung bei der Analyse von archivierten Messkurven oder der Beobachtung von online Trends ist der Vergleich zwischen Soll- und Istkurve. Sollkurven sind oftmals für bestimmte Situationen in verfahrenstechnischen Prozessen berechnet und geben Zielpunkte oder Kurvenverläufe für das Wartenpersonal vor.

Damit Sollkurven an aktuelle Gegebenheiten angepasst werden können, müssen sie innerhalb des Diagramms verschoben und ihre Stützstellen müssen verändert werden können. Jede analoge Kurve läßt sich für die interaktive Bearbeitung freischalten.

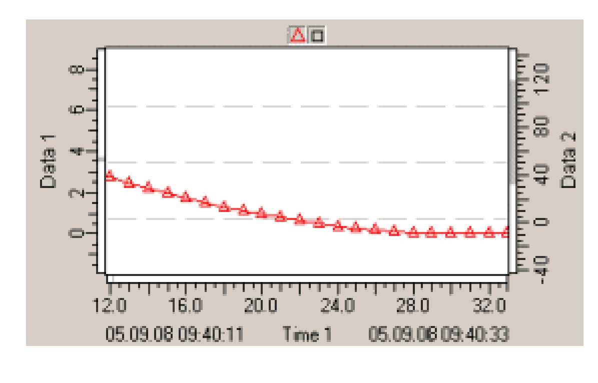
Hinweis: Alle interaktiven Veränderungen der Kurven verändern die Kurvenwerte, die in den *TrendViewer* geladen wurden. Sollen die Kurvenwerte aus dem Datenpuffer des *TrendViewer* ausgelesen werden, ist dieses zu beachten.

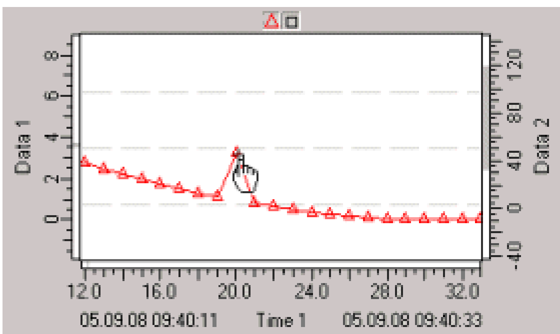
In der folgenden Abbildung wird die schwarze Sollkurve verschoben.





Die Stützstellen der roten Kurve werden mit der Maus verändert





Eine Applikation kann eine analoge Kurve auch über die Attribute **Offset** und **Gain** in x- und y-Richtung verschieben oder spreizen. Diese Veränderung beeinflusst die Kurvenwerte im Datenpuffer des *TrendViewer* nicht und kann jederzeit wieder zurückgesetzt werden.



5. TrendViewer Objekte

5.1 Eigenschaften

Die folgende tabellarische Darstellung enthält eine Übersicht über alle Objekte der Trend Komponente und deren Eigenschaften:

Objekt-Gruppen:



5.1.1 TrendCtrl

Allgemeine Eigenschaften der Trend-Komponente:

AutoScaleGrid	automatische Skalierung der Skalengrids (für `TRUE`)
AutoScaleVisibility	steuert für `TRUE` das automatische Ein- und Ausblenden der Skalen
BackColor	Hintergrundfarbe
Crosshair	Referenz zu einem TVCrosshairobjekt (**TVCros-shair**)
Curves	Die Anzahl der Kurven zurückliefern bzw. erhöhen
CurveButtons	Stil für die Kurvenknöpfe
DataWindows	aktiviert (`TRUE`) oder deaktiviert (`FALSE`) die Anzeige des Datenfensters beim Überstreichen der Skalen und Kurvenknöpfe
DockingScales	aktiviert (`TRUE`) oder deaktiviert (`FALSE`) die interaktive Verschiebbarkeit der Skalen
FixedFont	schaltet die automatische Anpassung der Schriftgröße aus (für `TRUE`)
Font	benutzte Schriftart (Änderung der Schriftart führt zu einer Neuberechnung der Größe (**FontScaleFactor**)
FontScaleFactor	Skalierungsfaktor für die Schriftgröße gegenüber der von der Trend-Komponente berechneten optimalen Darstellungsgröße
ForeColor	Vordergrundfarbe der Trend-Komponente
HorizontalScText	erzeugt für `TRUE` nur horizontale Beschriftungen für die senkrecht stehenden Skalen
HorizontalScTextAlign	Ausrichtung des horizontalen Skalentextes
Interactive	aktiviert (`TRUE`) oder deaktiviert (`FALSE`) die Maussteuerung
LockUpdate	„friert“ (für `TRUE`) das aktuelle Aussehen der Komponente ein
Orientation	Orientierung des Trend-Fensters an den Skalen bzw. der Skalen an das Trend-Fenster
OVBarsBackColor	Hintergrundfarbe der Übersichtsbalken
OVBarsRangeColor	Hintergrundfarbe des Bereichs der Übersichtsbalken
OverviewBars	Anzeigen (`TRUE`) oder Verstecken (`FALSE`) der Übersichtsbalken
ReleaseKey	Freigabetaste zur Positionierung der Skalen über die Mausbedienung
Ruler	Referenz zu einem TVRulerobjekt (**TVRuler**)
Rotated	Orientierung der Trendgrafik (`FALSE`: horizontal, `TRUE`: vertikal)
ScaleLegends	Anzeigen (`TRUE`) oder Verstecken (`FALSE`) der Skalenlegenden
ScaleSizeButton	Button für das Einstellen der Skalengröße
ScaleWidthFactor	Vergrößerungsfaktor für die Skalenbreite, um die Anzeige von waagerechten Skalenbeschriftungen im Text zu ermöglichen (s. **HorizontalScText**)
StripChartsStyle	aktiviert und konfiguriert die Stripchart-Funktionalität



SystemTimeZones	String-Array mit allen verfügbaren Zeitzonen
TrendAuxLineCount	Anzahl der Trend-Hilfslinien des Trend-Fensters
TrendAuxLineStart	Anfangswert für die Trend-Hilfslinien des Trend-Fensters
TrendAuxLineStep	Schrittweite für die Trend-Hilfslinien in Prozent des Trend-Fensters
TrendAuxLineStyle	Referenz zu einem **TVLineStyle**-Objekt. (Definiert den Stil einer Hilfslinie)
TrendBackColor	Farbe des Trend-Fenster-Hintergrunds
TrendPicture	Hintergrundbild für das Trend-Fenster
YScales	Die Anzahl der y-Skalen zurückliefern bzw. erhöhen
Zoom	erlaubt (`TRUE`) oder sperrt (`FALSE`) das Zoomen im Trend-FensterTrend-Fenster

AutoScaleGrid

`Boolean` AutoScaleGrid

Beschreibung

Diese Eigenschaft dient zum Ein- und Ausschalten der automatischen Skalierung der Teilungslinien bei den Skalen.

Wertebereich

`TRUE`	automatisches Skalierung ist eingeschaltet
`FALSE`	automatisches Skalierung ist ausgeschaltet

Anmerkungen

Soll eine anwenderspezifische Skalierung erfolgen, so muß diese Eigenschaft = `FALSE` gewählt werden. Die entsprechenden Einstellungen müssen dann in den Skalenobjekten über die Eigenschaften **DivGrid**, **MainGrid**, **SecGrid**, **GridTextFormat** vorgenommen werden.

Eine genauere Erläuterung dieser Eigenschaft findet sich innerhalb der Beschreibung von **MainGrid (TVXScale)**.

Siehe auch

TVXScale, TVYScale

AutoScaleVisibility

`Boolean` AutoScaleVisibility

Beschreibung

Diese Eigenschaft dient zum Ein- und Ausschalten des automatischen Managements für die Skalensichtbarkeit.

Wertebereich

`TRUE`	automatisches Skalenmanagement ist eingeschaltet
`FALSE`	automatisches Skalenmanagement ist ausgeschaltet



Anmerkungen

In der Voreinstellung sind innerhalb der Trend-Komponente nur die Skalen sichtbar, denen eine sichtbare Kurve zugeordnet wird. Skalen, denen keine sichtbaren Kurven zugeordnet sind, werden automatisch ausgeblendet.

Wird dieser Automatismus ausgeschaltet, ist die Sichtbarkeit der Skalen nicht mehr vom Zustand der Kurven abhängig und kann dann vom Container selbst gesteuert werden.

Siehe auch

TVXScale, TVYScale, TVCurve

BackColor

`System.Drawing.Color` BackColor

Beschreibung

Hintergrundfarbe der Trend-Komponente.

Wertebereich

wird durch die Windowsfarbpalette vorgegeben

Siehe auch

ForeColor

Crosshair

`TVCrosshair` Crosshair

Beschreibung

Referenz zu einem **TVCrosshair-**Objekt.

Wertebereich

siehe **TVCrosshair**

Siehe auch

Ruler

Curves

`Integer` Curves()

Beschreibung

Liefert die Anzahl der erzeugten **TVCurve**-Objekte zurück bzw. erhöht diese.



Wertebereich

`0..512`

Anmerkung

Eine Erhöhung der Anzahl an Kurven bis zur max. Grenze ist jederzeit möglich, eine Dezimierung dagegen ist nicht möglich !

Siehe auch

TVCurve

CurveButtons

`enumCurveButtons` CurveButtons

Beschreibung

Die Schaltknöpfe zum Ein- und Ausschalten der Kurven können mit Hilfe dieser Eigenschaft an bestimmten Positionen im Trend-Fenster plaziert werden:

Wertebereich

`0-cbNone`	Schaltknöpfe sind nicht sichtbar
`1-cbTop`	Schaltknöpfe an der Oberseite des Trend-Fensters
`2-cbScale`	Schaltknöpfe in den Skalen-Objekten

Anmerkungen

Die Schaltknöpfe dienen sowohl zum Ein- und Ausschalten der einzelnen Kurven als auch zur Anzeige von Name und Status jeder Kurve.

Anzeige:
Ein mit der Hintergrundfarbe ausgefüllter Knopf zeigt, daß die entsprechende Kurve sichtbar ist, ansonsten unsichtbar.

Wird der Maus-Zeiger über einen der Kurvenknöpfe bewegt, erscheint (für **DataWindows** = `TRUE`) ein kleines Fenster neben der Maus, welches den Namen der Kurve und die der Kurve zugeordnete Skala anzeigt.

Bedienung:
Durch einen einfachen Klick mit der linken Maustaste auf einen Kurvenknopf läßt sich die entsprechende Kurve sichtbar bzw. unsichtbar schalten.
Ist **CurveButtons** gleich `cbScales`, sollte das automatische Skalenmanagement (siehe **AutoScaleVisibility**) ausgeschaltet werden, da mit Ausschalten der Kurve auch die jeweilige Skala samt Knopf ausgeblendet wird.

Siehe auch

AutoScaleVisibility, **DataWindows**

DataWindows

`Boolean` DataWindows

Beschreibung

Definiert, ob die Datenfenster zur Anzeige von Trend-Daten, wie Skalen-Werte und Kurven-Beschreibungen, angezeigt werden sollen.



Wertebereich

TRUE	Datenfenster sind sichtbar
FALSE	keine Datenfenster.

Anmerkung

Das Datenfenster entspricht einem Windows Tooltip-Fenster, das der Bewegung des Mauszeigers folgt und Informationen zur aktuellen Mausposition anzeigt.

Siehe auch

CurveButtons

DockingScales

Boolean DockingScales

Beschreibung

Legt fest, ob die interaktive Positionierung der Skalen mit Hilfe der Maus aktiviert bzw. deaktiviert werden soll.

Bei Aktivierung werden die entsprechenden Bedienelemente in den Skalen sichtbar.

Wertebereich

TRUE	Skalenpositionierung mittels Mausbedienung wird aktiviert
FALSE	Skalenpositionierung mittels Mausbedienung wird deaktiviert

Siehe auch

TVXScale, TVYScale, ReleaseKey

FixedFont

Boolean FixedFont

Beschreibung

Schaltet die automatische Anpassung der Schriftgröße aus.

Wertebereich

TRUE	automatische Schriftgrößenanpassung ist ausgeschaltet
FALSE	automatische Schriftgrößenanpassung ist eingeschaltet

Anmerkung

Näheres zur automatischen Schriftgrößenanpassung siehe **Font**

Siehe auch

Font, FontScaleFactor



Font

`System.Drawing.Font` Font

Beschreibung

Definiert die verwendete Schriftart.

Wertebereich

wird durch die vorhandene Arbeitsumgebung vorgegeben

Anmerkungen

Um eine flexible Skalierbarkeit der Trend-Komponente zu erreichen, kann (für **FixedFont** gleich `FALSE`) die Größe der verwendeten Schrift automatisch an die Größe der Trend Komponente angepaßt werden. Dies geschieht nach dem folgenden Verfahren:

> Bei jedem Zeichenvorgang der Komponente wird die optimale Schriftgröße für die aktuelle Einstellung (abhängig von Skalen-Anzahl und Skalen-Positionen) berechnet. Die so errechnete Schriftgröße kann nun über die Eigenschaft **FontScaleFactor** vergrößert bzw. verkleinert werden, um die Schrift den Wünschen des Programmierers anzupassen.
>
> Bei jeder Änderung der Eigenschaft **Font** versucht die Komponente, die übergebene Schriftgröße durch Anpassung von **FontScaleFactor** einzustellen. Um das Aussehen der Komponente gegen Änderungen zu schützen, werden für die Schriftgröße Grenzwerte berechnet, die nicht überschritten werden können. Daher wird zwar die Eigenschaft **FontScaleFactor** exakt für die gewünschte Schriftgröße berechnet, jedoch kann die gewählte Schrift u. U. durch die Grenzwerte für die Schriftgröße tatsächlich etwas kleiner dargestellt werden.
>
> Nachdem die Eigenschaft **FontScaleFactor** intern angepaßt wurde, ist die Schriftgröße des aktuellen Fonts nicht mehr relevant für die Darstellung der Schrift. Die Abfrage des aktuellen Fonts liefert daher nicht die aktuelle Schriftgröße.
>
> Um einen Schrifttyp zu ändern, ohne die aktuelle Größe zu beeinflussen, sollte entsprechend dem folgenden Programmbeispiel verfahren werden:
>
> **Programm-Beispiel**

```
TrendViewer1.Font.Bold = True
TrendViewer1.FontScaleFactor = 2
```

Ist **FixedFont** gleich `TRUE`, wird die Schriftart durch die Trend Komponente nicht geändert.

Siehe auch

FontScaleFactor, **HorizontalScText**, **FixedFont**, **ScaleWidthFactor**



FontScaleFactor

`Double` FontScaleFactor

Beschreibung

Vergrößerungs- bzw. Verkeinerungsfaktor für die berechnete optimale Font-größe.

Wertebereich

`>0 und <1.0`	verkleinern der Schrift
`1.0`	keine Änderung der Schriftgröße
`>1.0`	vergrößern der Schrift

Anmerkungen und Programm-Beispiel

Diese Eigenschaft ist nur für (**FixedFont** gleich `FALSE`) relevant.

Näheres zur automatischen Größenanpassung der Schriftart siehe Eigenschaft **Font**.

Siehe auch

Font, **HorizontalScText**, **FixedFont**

ForeColor

`System.Drawing.Color` ForeColor

Beschreibung

Definiert die Vordergrundfarbe der Trend-Komponente.

Wertebereich

wird durch die Windowsfarbpalette vorgegeben

Anmerkung

Die Vordergrundfarbe dient zum Zeichnen der Umrandung des Trend-Fenster sowie als Standard-Schriftfarbe für die Datenfenster.

Siehe auch

BackColor, **DataWindows**

HorizontalScText

`Boolean` HorizontalScText

Beschreibung

Horizontale Beschriftung von senkrecht stehenden Skalen.

Wertebereich

`TRUE`	alle Skalen werden horizontal beschriftet
`FALSE`	die Skalen werden entsprechend ihrer Ausrichtung beschriftet



Anmerkungen

In der Voreinstellung der Trend-Komponente ist **HorizontalScText** auf `FALSE` gesetzt. Alle Skalen werden dann entsprechend ihrer Ausrichtung beschriftet. Dies erlaubt eine platzsparende Darstellung der Trend-Komponente.

Durch das Setzen von **HorizontalScText** auf `TRUE` kann erreicht werden, daß alle Skalentexte (mit Ausnahme der Skalen-Bezeichner) horizontal angezeigt werden. Diese Art der Skalenbeschriftung benötigt aber eine veränderliche Breite der Skalen, um den entsprechenden Text darstellen zu können. Die Breite der Skalen muß dann vom Anwender oder Container entsprechend angepaßt werden. Zur Unterstützung von variablen Skalenbreiten können die Eigenschaft **ScaleWidthFactor** verwendet werden. Das Management für die Skalenbreite wird erheblich vereinfacht, wenn die automatische Schriftgrößenanpassung (s.**FixedFont**) ausgeschaltet ist.

Siehe auch

Font, **FontScaleFactor**, **ScaleWidthFactor**, **FixedFont**

HorizontalScTextAlign

`Integer` HorizontalScTextAlign

Beschreibung

Ausrichtung des Textes bei horizontaler Beschriftung.

Wertebereich

`0`	Ausrichtung an der Teilungslinie
`1`	Ausrichtung links/rechtsbündig zum Textrand

Siehe auch

Font, **HorizontalScText**, **FixedFont**

Interactive

`Boolean` Interactive

Beschreibung

Schaltet für `FALSE` alle Maus-Interaktionen ab. Nur der Aufruf des Benutzerdialogs durch Doppelklick der linken Maustaste wird durch **Interactive** nicht beeinflußt.

Wertebereich

`TRUE`	Maussteuerung ist aktiviert
`FALSE`	Maussteuerung ist abgeschaltet

Siehe auch

Interactive (TVXScale), **Interactive (TVYScale)**



LockUpdate

`Boolean` LockUpdate

Beschreibung

Ist **Lockupdate** auf `TRUE` gesetzt, wird das Neuzeichnen der Komponente gesperrt. Änderungen in den Trend-Eigenschaften sind in der Komponente nicht mehr sichtbar.

Wertebereich

`TRUE`	Änderungen in den Eigenschaften haben keine Auswirkung auf das Aussehen der Komponente
`FALSE`	Änderungen in den Eigenschaften werden sofort sichtbar, Voreinstellung

Anmerkungen

Bevor **LockUpdate** auf `TRUE` gesetzt wird, sollte die Komponente zumindest einmal vollständig gezeichnet worden sein.

Eine Änderung der Fenstergröße der Komponente während **LockUpdate** auf `TRUE` gesetzt ist, wird ebenfalls keine Auswirkung auf die Darstellung haben. Deshalb sollte zur Änderung der Komponenten-Größe stets **LockUpdate** auf `FALSE` gesetzt werden.

Wird **LockUpdate** auf `TRUE` gesetzt und läuft ein Hintergrundprozeß zum Zeichnen der Komponente, so wird gewartet, bis der Hintergrundprozeß abgeschlossen wurde

Programm-Beispiel

siehe **ActualMax, ActualMin, ActualMaxDateTime, ActualMinDateTime**

Siehe auch

TVRange

Orientation

`Short` Orientation

Beschreibung

Legt den Darstellungsmodus des Kurvenfensters fest.

Wertebereich

`0`	Kurvenfenster und Skalen werden automatisch der Gesamtgröße des Trend-Fensters angepaßt
`1`	fixes Kurvenfenster. Skalen werden angepaßt

Anmerkungen

Bei einem fixen Kurvenfenster wird der kleinste Abstand zwischen Kurvenfenster und Trendrand als Basisbreite/höhe für die Skalen genommen.

Eine genauere Erläuterung dieser Eigenschaft findet sich innerhalb der Beschreibung von **SetMarginFactor**.



Siehe auch

TVXScale, TVYScale, SetMarginFactor

OVBarsBackColor

`System.Drawing.Color` OVBarsBackColor

Beschreibung

Hintergrundfarbe des Balkens, der das Datenintervall (Datenskala) oder das Zeitintervall (Zeitskala) anzeigt.

Wertebereich

wird durch die Windowsfarbpalette vorgegeben

Siehe auch

OVBarsRangeColor, OverviewBars

OVBarsRangeColor

`System.Drawing.Color` OVBarsRangeColor

Beschreibung

Hintergrundfarbe des Bereichs für die Übersichtsbalken.

Wertebereich

wird durch die Windowsfarbpalette vorgegeben

Siehe auch

OVBarsBackColor, **OverviewBars**

OverviewBars

`Boolean` OverviewBars

Beschreibung

Schaltet die Übersichtsbalken der Skalen ein (`TRUE`) bzw. aus (`FALSE`).

Wertebereich

`TRUE`	die Übersichtsbalken sind sichtbar
`FALSE`	die Übersichtsbalken sind versteckt

Anmerkung

Fällt die Schriftgröße unterhalb eines bestimmten Grenzwerts, werden die Übersichtsbalken automatisch ausgeblendet.

Siehe auch

OVBarsBackColor, **OVBarsRangeColor**



ReleaseKey

`Boolean` ReleaseKey

Beschreibung

Freigabetaste ein (`TRUE`) bzw. ausschalten (`FALSE`).

Wertebereich

`TRUE`	Freigabetaste aktiviert
`FALSE`	Freigabetaste deaktiviert

Anmerkungen
Die Freigabetaste ermöglicht es, die Skalen interaktiv mit Hilfe der Maus zu Positionieren.

Diese Eigenschaft und Funktionalität ist je nach Lizenzmodell eingeschränkt bzw. nicht verfügbar !

Siehe auch

DockingScales

Ruler

`TVRuler` Ruler

Beschreibung

Referenz zu einem **TVRuler-**Objekt.(Trendlineal)

Wertebereich

siehe **TVRuler**

Rotated

`Boolean` Rotated

Beschreibung

Im allgemeinen ist die x-Skala eines Trends horizontal mit dem Skalenminimum links und dem Skalenmaximum rechts orientiert.

In besonderen Fällen ist eine vertikale Orientierung der x-Skala mit dem Skalen-minimum unten gewünscht. Diese Art des Trends kann durch Setzen von **Rotated** auf `TRUE` erreicht werden.

Wertebereich

`TRUE`	vertikale x-Skala
`FALSE`	horizontale x-Skala (standard)



ScaleLegends

`Boolean` ScaleLegends

Beschreibung

Schaltet die Skalenlegenden ein (`TRUE`) bzw. aus (`FALSE`).

Wertebereich

`TRUE`	die Legenden sind sichtbar
`FALSE`	die Legenden sind versteckt

Siehe auch

OverviewBars

ScaleSizeButton

`Boolean` ScaleSizeButton

Beschreibung

Button, um die Größe einer Skala bei gestapelten Skalen einzustellen.

Wertebereich

`TRUE`	die Buttons sind sichtbar
`FALSE`	die Buttons sind versteckt

Siehe auch

DockingScales

ScaleWidthFactor

`Double` ScaleWidthFactor

Beschreibung

Ermöglicht die Verbreiterung aller senkrecht stehenden Skalen um den angegebenen Faktor.

Wertebereich

alle positiven Werte

Anmerkungen

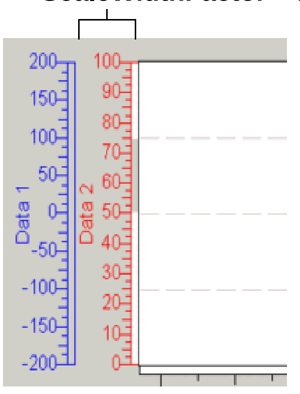
Diese Eigenschaft wird benötigt, um die Darstellung von horizontalen Skalenbeschriftungen an senkrechten Skalen zu ermöglichen.

Vor jedem Zeichenvorgang berechnet die Trend-Komponente Position und Größe aller Trend-Elemente, dabei wird ebenfalls die Breite aller Skalen berechnet. Die Skalenbreite aller senkrecht stehenden Skalen wird anschließend mit dem Wert von **ScaleWidthFactor** multipliziert.

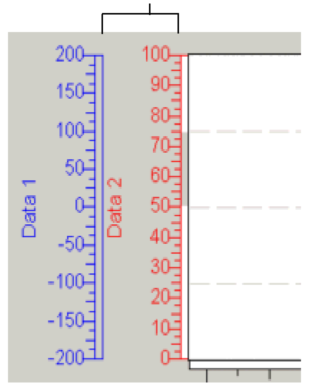
Die folgende Abbildungen veranschaulichen die Bedeutung dieser Eigenschaft:



ScaleWidthFactor = 1.0



ScaleWidthFactor = 1.3



In dem oben gezeigten Beispiel wird der Wert von **ScaleWidthFactor** von 1.0 auf 1.3 erhöht, um eine korrekte Anzeige der Skalenbeschriftung zu ermögli-chen.

Siehe auch

Font, **FontScaleFactor**, **HorizontalScText**

StripChartsStyle

`enumStripChartsStyle` StripChartsStyle

Beschreibung

Mit Hilfe dieser Eigenschaft, kann die Trend-Komponente zur Darstellung von Kurvenbändern (Stripcharts) umkonfiguriert werden.

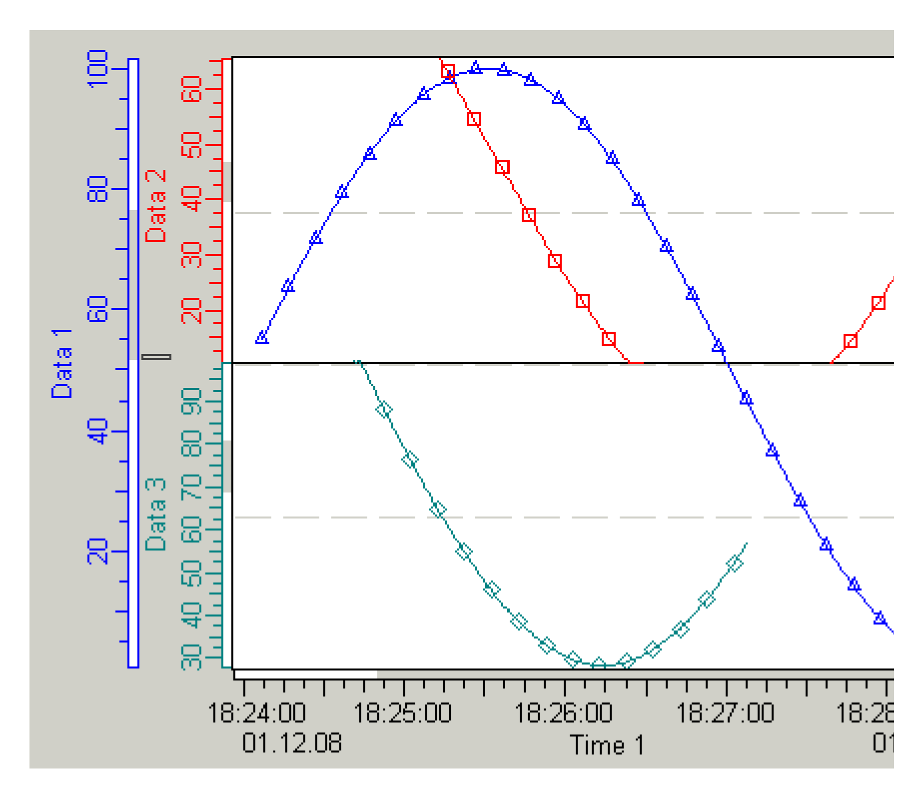
Wertebereich

`0-scNoStripCharts`	keine Stripcharts
`1-scLeftSide`	Stripcharts nur am linken Rand
`2-scRigthSide`	Stripcharts nur am rechten Rand
`3-lsBothSides`	Stripcharts an beiden Rändern



Anmerkungen

Wird die Darstellung von Kurvenbändern aktiviert, so wird das Trend-Fenster in Bänder aufgeteilt. Diese Bänder orientieren sich an der Aufteilung des jeweiligen Trendrands durch die Skalen. Es werden dabei nur jene Skalen berücksichtigt, die direkt am Rand des Trend-Fensters plaziert sind. Kurven, die diesen Skalen zugeordnet sind, werden sobald Sie das jeweilige Band verlassen, abgeschnitten. Die folgende Abbildung erläutert anhand eines konkreten Beispiels die Kurvenbänder der Trend-Komponente für **StripChartsStyle** gleich `scLeftSide`:



Im oben gezeigten Beispiel bilden die Skalen „Data 2“ und „Data 3“ die Bänder und die zu diesen Skalen gehörenden Kurven werden abgeschnitten, sobald sie die Grenzen der Bänder überschreiten. Da die zur Skala „Data 1“ gehörende Kurve keinem Band zugeordnet ist, wird sie im gesamten Trend-Fenster dargestellt.

SystemTimeZones

`String[]` SystemTimeZones

Beschreibung

Liefert ein String-Array mit allen verfügbaren Zeitzonen des TrendViewers zurück.

Eine Zeitzone findet u.a. Verwendung in der Eigenschaft "TimeZoneName" der **TVData** und **TVXScale** Objekten.



Siehe auch

TVData, TVXScale

TrendAuxLineCount

`Short` TrendAuxLineCount

Beschreibung

Legt die Anzahl der Hilfslinien fest.

Wertebereich

beliebiger ganzzahliger Wert

Siehe auch

TrendAuxLineStart, **TrendAuxLineStep, TrendAuxLineStyle**

TrendAuxLineStart

`Short` TrendAuxLineStart

Beschreibung

Legt die Startposition der ersten Hilfslinie fest.

Wertebereich

beliebiger Wert

Siehe auch

TrendAuxLineStep, **TrendAuxLineStyle, TrendAuxLineCount**

TrendAuxLineStep

`Short` TrendAuxLineStep

Beschreibung

Definiert die Schrittweite der Unterteilung des Trend-Fensters durch die Hilfslinien in Prozent zur Höhe des Trend-Fensters.

Wertebereich

`<=0%`	keine Hilfslinien
`1% - 99%`	Abstand der Hilfslinien untereinander (Vorgabe `25`%)
`>=100%`	keine Hilfslinien

Anmerkung

Für **TrendAuxLinesStep** gleich `10`% wird das Trend-Fenster durch die Hilfslinien in 10 Bereiche unterteilt, für **TrendAuxLineStep** gleich `25`% wird das Trend-Fenster geviertelt.

Siehe auch

TrendAuxLineStyle, TrendAuxLineCount, TrendAuxLineStart



TrendAuxLineStyle

`TVLineStyle` TrendAuxLineStyle

Referenz zu einem **TVLineStyle**-Objekt. Legt den Linienstil der Trend-Hilfslinien fest.

Wertebereich

siehe **TVLineStyle**

Anmerkung

Das Zeichnen von Linien, deren Stil nicht gleich `psSolid` und deren Linienbreite nicht gleich `0` ist, nimmt erheblich mehr Zeit in Anspruch.

Siehe auch

TrendAuxLineStep, TrendAuxLineCount, TrendAuxLineStart

TrendBackColor

`System.Drawing.Color` TrendBackColor

Beschreibung

Hintergrundfarbe des Trend-Fensters.

Wertebereich

wird durch die Windowsfarbpalette vorgegeben

Siehe auch

BackColor

TrendPicture

`System.Drawing.Bitmap` TrendPicture

Beschreibung

Hintergrundbild des Trend-Fensters.

Wertebereich

jedes Windows-Bitmap

Anmerkung

DieTrend-Komponente übernimmt kein eigenes Farbmanagement, um die Palette des gewählten Trend-Hintergrundbildes zu realisieren. Dies muß die Applikation übernehmen.

Siehe auch

GetXScale



YScales

`Integer` YScales()

Beschreibung

Liefert die Anzahl der erzeugten y-Skalen-Objekte zurück bzw. erhöht diese.

Wertebereich

`0..512`	`Anzahl der erzeugten y-Skalen`

Anmerkung

Eine Erhöhung der Anzahl an Y-Skalen bis zur max. Grenze ist jederzeit möglich, einer Dezimierung dagegen ist nicht möglich !

Siehe auch

TVYScale

Zoom

`enumTrendZoom` Zoom

Beschreibung

Stellt die Zoom-Methode innerhalb des Trend-Fensters ein.

Wertebereich

`siehe` **enumTrendZoom**



5.1.2 TVXScale

Eigenschaften des x-Skalenobjekts :

AbsoluteTime	schaltet die Zeitskala zwischen Uhrzeit (`TRUE`) und Zeitdauer (`FALSE`) um (s.**Type**)
AutoFollowMode	aktiviert (`TRUE`) oder deaktiviert (`FALSE`) das automatische Umschalten in den Folgemodus
AuxCount	Anzahl der Hilfslinien
AuxLine	Definiert die Anordnung der Hilfslinien
AuxLineStyle	Referenz zu einem **TVLineStyle-**Objekt
AuxStart	Startpunkt der ersten Hilfslinie
AuxStep	Schrittweite der Hilfslinien in Prozent
DataRangeEnabled	zeigt (`TRUE`) oder versteckt (`FALSE`) die Anzeige des aktuellen Wertebereichs innerhalb des Übersichtbalkens
DivGrid	Definiert das Teilungsraster. Das Teilungsraster legt die Anzahl der Unterteilungen innerhalb des Nebenrasters (**SecGrid**) fest
EdgeTextFormat	Format für die Textausgabe links/rechts in der Legende
GridTextFormat	Textformat der Werteanzeige an den Hauptteilungslinien
Index	Instanz der Skala
FollowMode	aktiviert (`TRUE`) oder deaktiviert (`FALSE`) den Folgemodus
FollowModeButton	zeigt (`TRUE`) oder versteckt (`FALSE`) den Knopf für den Folgemodus an der Skala
ForeColor	Vordergrundfarbe der Skala
Enabled	aktiviert (`TRUE`) oder deaktiviert (`FALSE`) die Skala (änderbar nur für **AutoScaleVisibility**=`FALSE`)
Index	Instanz der Skala
Interactive	aktiviert (`TRUE`) oder deaktiviert (`FALSE`) die Mausbedienung
LegendDateVisible	erweiterte Datumsanzeige in der Legende der Skala ein (`TRUE`) - bzw. ausschalten (`FALSE`)
MainGrid	Definiert das Hauptraster. Ein Hauptraster ist der Abstand zwischen zwei Hauptteilungslinien
Mirrored	Spiegelt die Skala
Modulo360Degrees	zeigt bei einer Gradskala nur Werte zwischen 0° und 360° (bildet immer positives Modulo 360°)
Position	Referenz zu einem **TVPosition-**Objekt (Positionierung der Skala)
Priority	Legt die Priorität der Skala fest
Range	Referenz zu einem **TVRange-**Objekt (Datenbereich der Skala)
RangeMoveOnly	gestattet für `TRUE` nur die Verschiebung des Wertebereichs
Res	maximale Anzeigegenauigkeit (nicht persistent für Zeitachsen)



SecGrid	Definiert das Nebenraster. Das Nebenraster legt Anzahl der Unterteilungen innerhalb des Hauptrasters (**MainGrid**) fest
Shared	Legt fest, ob alle Kurven die selben x-Daten benutzen (TRUE) oder ob jede Kurve einen eigenen x-Datensatz benötigt
Text	Skalentext zur Beschreibung der x-Skala
TextAlign	Textausrichtung des Skalentextes in der Legende
TimeZoneName	Legt die Zeitzone für die Skala fest
ToolTipFormat	Format für die Text-u.Werteausgabe im Tooltip (Datenfenster)
Type	bestimmt das Anzeigeformat der Skala
Visible	zeigt (`TRUE`) oder versteckt (`FALSE`) die Skala (**AutoScaleVisibility**=`FALSE`)

AbsoluteTime

`Boolean` AbsoluteTime()

Eine Zeitskala wird entweder mit Datum und Uhrzeit dargestellt oder mit einer Zeitspanne.

Wertebereich

`FALSE`	Zeitspanne (Voreinstellung)
`TRUE`	Absolute Zeitangabe

Anmerkungen

Die Beschriftung der Skala zeigt bei der absoluten Zeitangabe entweder das Datum oder die Uhrzeit. Zeitspannen werden in Sekunden, Minuten, Stunden und Tagen angezeigt.

Der Skalennullpunkt zur Anzeige von Zeitspannen wird durch **DataMax, DataMin, DataMaxDateTime, DataMinDateTime** festgelegt. Um das interaktive Umschalten zwischen Absolutzeit und Zeitspanne zu ermöglichen, müssen alle Wertebereiche (siehe **ActualMax, ActualMin, ActualMaxDateTime, ActualMinDateTime, DataMax, DataMin, DataMaxDateTime, DataMinDateTime**) in absoluten Zeitwerten angegeben werden.

Diese Eigenschaft ist nur für Zeitskalen mit **Type** = `stTime` relevant.

Siehe auch

ActualMax, ActualMin, ActualMaxDateTime, ActualMinDateTime, DataMax, DataMin, DataMaxDateTime, DataMinDateTime, **Res**

AutoFollowMode

`Boolean` AutoFollowMode()

Beschreibung

Schaltet den automatischen Folgemodus (s. **FollowMode**) ein bzw. aus.



Wertebereich

`TRUE`	schaltet automatisch den Folgemodus ein
`FALSE`	keine automatische Einschaltung des Folgemodus

Anmerkung

Ist Autofollowmode gleich `TRUE`, wird der Folgemodus aktiviert, sobald das Maximum der x-Werte das Maximum des dargestellten Bereichs erreicht.

Siehe auch

FollowMode, **FollowModeButton**

AuxCount

`Integer` AuxCount()

Beschreibung

Definiert die Anzahl von Hilfslinien, die gezeichnet werden sollen.

Wertebereich

`-1`	unbegrenzte Anzahl von Hilfslinien (z.B. für kontinuierliche Hilfslinienanzeige)
`<=0`	keine Hilfslinien
`>0`	Anzahl gezeichneter Hilfslinien

Anmerkungen

Bei der Erzeugung von x-Skalen-Hilfslinien werden ab der Position **AuxStart** die durch **AuxCount** bestimmte Anzahl von Hilfslinien mit der Schrittweite **AuxStep** gezeichnet.

Eine genauere Erläuterung dieser Eigenschaft findet sich innerhalb der Beschreibung von **AuxLine**.

Siehe auch

AuxStart, **AuxStep**, **AuxLineStyle, AuxLine**

AuxLine

`enumAuxLineTyp` AuxLine()

Beschreibung

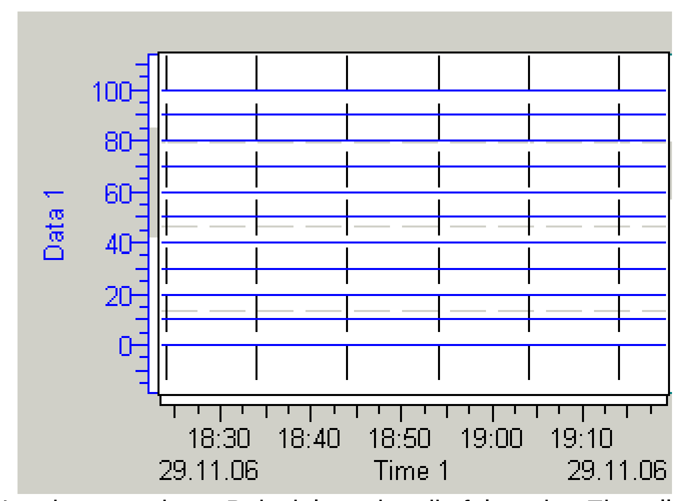
Definiert die Anordnung der Hilfslinien.



Wertebereich

`tlClassic`	Hilfslinien werden über die Anzahl der Linien (**AuxCount**), Start (**AuxStart**) und Step (**AuxStep**) definiert
`tlGrid`	Hilfslinien werden über **AuxStep** an den Teilungslinien (**DivGrid, MainGrid, SecGrid**) der Skalen ausgerichtet)

Beispiel für AuxLine = 0 an der x-u.y-Skala:



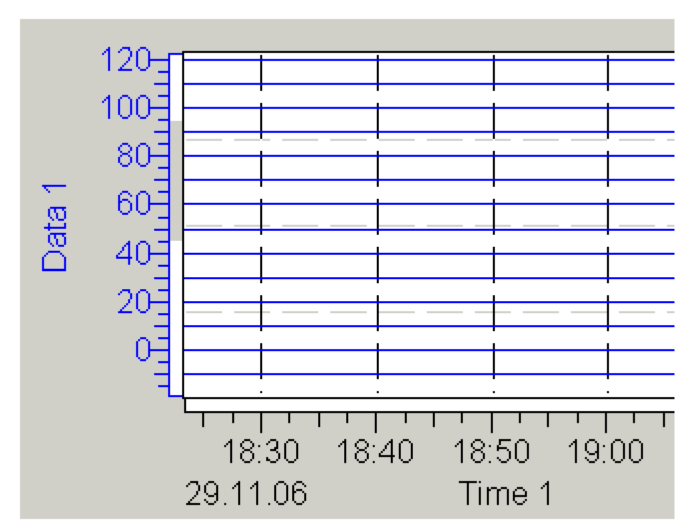
Im oben gezeigten Beispiel wurden die folgenden Einstellungen vorgenommen:

Einstellung der x-Skala-Hilfslinien auf kontinuierliche Rasterung (alle 10min):
AuxStart() = 0.0
AuxStep() = 600.0
AuxCount() = -1
AuxLineStyle() = `psDash`

Einstellung der y-Skala-Hilfslinien mit:
AuxStart(`dsYScale1`) = 0.0
AuxStep(`dsYScale1`) = 10.0
AuxCount(`dsYScale1`) = 11



Beispiel für AuxLine = 0 an der x-Skala und **AuxLine** = 1 an der y-Skala:



Im oben gezeigten Beispiel wurden die folgenden Einstellungen vorgenommen:

Einstellung der x-Skala-Hilfslinien auf kontinuierliche Rasterung (alle 10min):
AuxStart() = 0.0
AuxStep() = 600.0
AuxCount() = -1
AuxLineStyle() = `psDash`
Einstellung der y-Skala-Hilfslinien mit:
AuxStart(`dsYScale1`) = 0.0 (nicht relevant)
AuxStep(`dsYScale1`) = 1 (Ausrichtung an **MainGrid** und **SecGrid**)
AuxCount(`dsYScale1`) = 0 (nicht relevant)

Anmerkung

Bei AuxLine = 1 hat die Eigenschaft **AuxStart** und **AuxCount** keine Bedeutung !

Siehe auch

AuxStart, **AuxStep**, **AuxLineStyle, AuxCount**

AuxLineStyle

`TVLineStyle` AuxLineStyle()

Beschreibung

Referenz auf ein **TVLineStyle**-Objekt. Legt den Linienstil der x-Skalen-Hilfslinien fest.

Wertebereich

siehe **TVLineStyle**

Siehe auch

AuxStart, **AuxStep**, **AuxLine, AuxCount**



AuxStart

`Double` AuxStart()

Beschreibung

Legt den Startwert für das Zeichnen der Hilfslinien fest.

Wertebereich

beliebiger Wert

Anmerkung

Eine genauere Erläuterung dieser Eigenschaft findet sich innerhalb der Beschreibung von **AuxLine**.

Siehe auch

AuxCount, AuxStep, AuxLine, AuxLineStyle

AuxStep

`Double` AuxStep(`enumXScales` ScaleNumber)

Beschreibung

Legt die Schrittweite zwischen den Hilfslinien fest. Die Schrittweite hat je nach **AuxLine**-Type einen unterschiedlichen Wertebereich.

Wertebereich

für **AuxLine** = 0:

beliebiger Wert

für **AuxLine** = 1:

`<0`	keine Hilfslinien
`0`	Ausrichtung der Hilflinien an **DivGrid, MainGrid, SecGrid**
`1`	Ausrichtung der Hilflinien an **MainGrid, SecGrid**
`2`	Ausrichtung der Hilflinien an **MainGrid**
`>2`	keine Hilfslinien

Anmerkung

Eine genauere Erläuterung dieser Eigenschaft findet sich innerhalb der Beschreibung von **AuxLine**.

Siehe auch

AuxCount, AuxStart, AuxLine, AuxLineStyle, DivGrid, MainGrid, SecGrid



DataRangeEnabled

`Boolean` DataRangeEnabled()

Beschreibung

Schaltet die Darstellung des aktuellen Datenintervalls im Übersichtsbalken ein. Das Datenintervall wird durch die Eigenschaften **DataMax, DataMin, DataMax-DateTime, DataMinDateTime** aus dem **TVRange**-Objekt festgelegt.

Wertebereich

`FALSE`	nicht eingeschaltet, Voreinstellung
`TRUE`	Datenintervall wird dargestellt

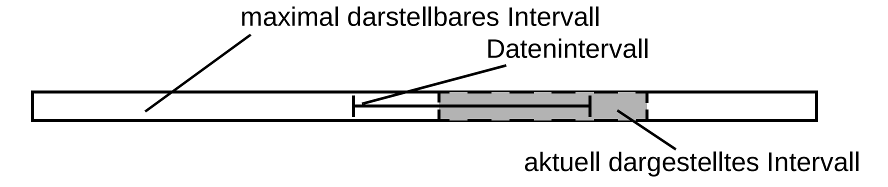
Anmerkungen

Bei der Darstellung von Meßwertkurven ist es hilfreich, den größten und den kleinsten Wert innerhalb der Meßwertmenge zu kennen, damit die Meßwert-kurve mit der besten Auflösung dargestellt werden kann. Außerdem hilft die Darstellung des Datenintervalls festzustellen, welcher Ausschnitt des Datenin-tervalls im Vergleich zur Skala dargestellt wird.

Der Datenbereich wird während der Laufzeit festgelegt. Zur Entwicklungszeit wird ein Standard Datenbereich eingestellt, um das spätere Aussehen der Skala anzudeuten.

Beispiel (Datenskala)

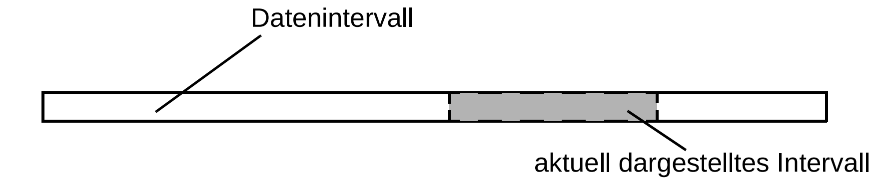
Der Zusammenhang zwischen dem maximal darstellbaren Intervall, dem aktuell dargestellten Intervall und dem Datenintervall geht aus folgender Skizze hervor:



Bei dieser Konstellation zeigt die Skala aktuell nur einen kleinen Ausschnitt aus der gesamten möglichen Grafik an. Von dem Datenintervall ist ebenfalls nur ein Ausschnitt zu sehen.

Beispiel (Zeitskala)

Da der maximal darstellbare Zeitbereich einer Zeitskala nicht verändert werden kann, wird hierfür der Übersichtsbalken etwas anders dargestellt, wie folgende Skizze zeigt:



Der Übersichtsbalken der Zeitskala entspricht der für diese Übersichtsbalken üblichen Darstellung.



Programm-Beispiel

```
Dim XScale As TrendViewer30.CTVXScale
Dim XRange As TrendViewer30.CTVRange

XScale = m_Trend.GetXScale(0)
XRange = XScale.Range

XScale.DataRangeEnabled = False
XRange.DataMin = 10
XRange.DataMax = 100
XScale.DataRangeEnabled = True
```

Siehe auch

DataMax, DataMin, DataMaxDateTime, DataMinDateTime

DivGrid

`Integer` DivGrid()

Beschreibung

Definiert das Teilungsraster. Das Teilungsraster legt Anzahl der Unterteilungen innerhalb des Nebenrasters (**SecGrid**) fest.

Wertebereich

beliebiger ganzzahliger Wert

Anmerkungen

Eine genauere Erläuterung dieser Eigenschaft findet sich innerhalb der Beschreibung von **MainGrid**.

DivGrid ist nur relevant, wenn AutoScaleGrid gleich FALSE ist, ansonsten werden die Einstellungen aus der XML-Formatdatei bzw. den Standardeinstellungen des *TrendViewers* übernommen.

Siehe auch

MainGrid, **SecGrid, AutoScaleGrid**

EdgeTextFormat

`String` EdgeTextFormat()

Beschreibung

Bestimmt das Format der Werteausgabe "links und rechts" in der Legende.



Wertebereich

Formatierungstyp in .Net-Notation z.B. "`{0:HH:mm:ss}`"

Anmerkung

EdgeTextfFormat ist nur relevant, wenn **AutoScaleGrid** gleich FALSE ist, ansonsten werden die Einstellungen aus der XML-Formatdatei bzw. den Standardeinstellungen des *TrendViewers* übernommen.

Siehe auch

AutoScaleGrid, LegendDateVisible

Enabled

`Boolean` Enabled()

Beschreibung

Aktiviert oder deaktiviert eine x-Skala.

Wertebereich

`TRUE`	Skala ist aktiviert
`FALSE`	Skala ist deaktiviert

Anmerkungen

Diese Eigenschaft kann nur geändert werden, wenn **AutoScaleVisibility** = `FALSE` ist, andernfalls übernimmt die Trend-Komponente die Aktivierung der Skalen selbst.

Es können nur x-Skalen sichtbar geschaltet (s.**Visible**) werden, die zuvor aktiviert wurden.

Siehe auch

AutoScaleVisibility, **Visible**

FollowMode

`Boolean` FollowMode()

Beschreibung

Legt fest, ob die Komponente im Folgemodus arbeitet.

Im Folgemodus wird der sichtbare Bereich der x-Achse immer so verschoben, daß die rechte Seite des Trend-Fensters dem aktuellen Wert des Maximums der x-Werte entspricht. Für fortlaufende Meßreihen sind so immer die aktuellsten Daten im Trend-Fenster sichtbar.

Wertebereich

`TRUE`	Folgemodus ist aktiv
`FALSE`	Folgemodus ist inaktiv



Anmerkungen

Diese Eigenschaft kann sich während der Laufzeit durch folgende Ereignisse ändern:

- der Anwender hat das Maximum der x-Skala geändert
- der Anwender hat den Folgemodus durch einen Maus-Klick auf den Bedienknopf für den Folgemodus ausgeschaltet

von `FALSE` nach `TRUE`:

- für **AutoFollowMode** gleich `TRUE` erreichte das Maximum der x-Daten den rechten Rand des Trend-Fensters
- der Anwender hat den Folgemodus durch einen Maus-Klick auf den Bedienknopf für den Folgemodus eingeschaltet

Um eine Änderung dieser Eigenschaft durch den Benutzer zu verhindern, muß die x-Skala durch Abschalten der interaktiven Bedienung (s. **Interactive**) statisch geschaltet werden und der Bedienknopf für den Folgemodus (s. **FollowModeButton**) ausgeblendet werden.

Siehe auch

AutoFollowMode, **FollowMode, FollowModeButton**, **Interactive**

FollowModeButton

`Boolean` FollowModeButtons

Beschreibung

Definiert, ob die Bedien- und Anzeigeknöpfe für den Folgemodus der x-Skalen angezeigt werden sollen.

Wertebereich

`TRUE`	Bedienknöpfe für den Folgemodus werden angezeigt
`FALSE`	Bedienknöpfe für den Folgemodus werden nicht angezeigt

Anmerkungen

Die Bedienknöpfe für den Folgemodus (für **FollowModeButtons** gleich `FALSE`) dienen sowohl der Anzeige als der Bedienung.

Anzeige:
Ist der Folgemodus einer x-Skala aktiv, wird der jeweilige Knopf gedrückt dargestellt. Das Symbol für den Folgemodus ist dann hellgrün eingefärbt.

Bedienung:
Durch einen einfachen Klick mit der linken Maustaste auf einen Bedienknopf läßt sich der Folgemodus der jeweiligen x-Skala ein- bzw. ausschalten.

Siehe auch

AutoFollowMode, FollowMode



ForeColor

`System.Drawing.Color` ForeColor()

Beschreibung

Skalen und Beschriftungsfarbe der x-Skala.

Wertebereich

wird durch die Windowsfarbpalette vorgegeben

Siehe auch

BackColor

GridTextFormat

`String` GridTextFormat()

Beschreibung

Bestimmt das Format der Werteanzeige für die Hauptraster.

Wertebereich

Formatierungstyp in C-Notation z.B. "`{0:HH:mm:ss}`"

Anmerkungen

Eine genauere Erläuterung dieser Eigenschaft findet sich innerhalb der Beschreibung von **MainGrid**.

GridTextFormat ist nur relevant, wenn AutoScaleGrid gleich FALSE ist, ansonsten werden die Einstellungen aus der XML-Formatdatei bzw. den Standardeinstellungen des *TrendViewers* übernommen.

MainGrid, AutoScaleGrid

Index

`Short` Index()

Beschreibung

Liefert den Indexwert der x-Skala.

Wertebereich

`0..MaxXScales`

Interactive

`Boolean` Interactive()

Beschreibung

Schaltet die interaktive Veränderung des Darstellungsbereichs der x-Skala mit Hilfe der Maus ein (`TRUE`) bzw. aus (`FALSE`).



Wertebereich

`TRUE`	Benutzer kann den Darstellungsbereich der x-Skala mit Hilfe der Maus verändern
`FALSE`	Änderung gesperrt

Anmerkung

Die Zoomfunktionen des Trend-Fensters haben bei **Interactive** = `FALSE` keine Auswirkung auf die x-Skala.

Siehe auch

Zoom (TrendCtrl)

LegendDateVisible

`Boolean` LegendDateVisible()

Beschreibung

Legt fest, ob die erweiterte Datumsanzeige der x-Skala in der Legende der Skala ein (`TRUE`) - bzw. ausgeschaltet (`FALSE`) ist.

Wertebereich

`TRUE`	erweiterte Datumsanzeige der xSkala wird links und rechts in der Legend angezeigt
`FALSE`	erweiterte Datumsanzeige wird nicht angezeigt

Anmerkungen

Nur relevant bei einer Zeitskala (tsTime). Wurde die Zeitskala z.B. soweit gezoomt, dass nur noch Stunden, Minuten und Sekunden dargestellt werden, kann in der erweiterten Datumsanzeige z.B. noch der Tag, Monat und Jahr dargestellt werden.

Siehe auch

EdgeTextFormat

MainGrid

`Double` DivGrid()

Beschreibung

Das Hauptraster definiert den Abstand zwischen zwei Hauptteilungslinien.

Wertebereich

beliebiger Wert

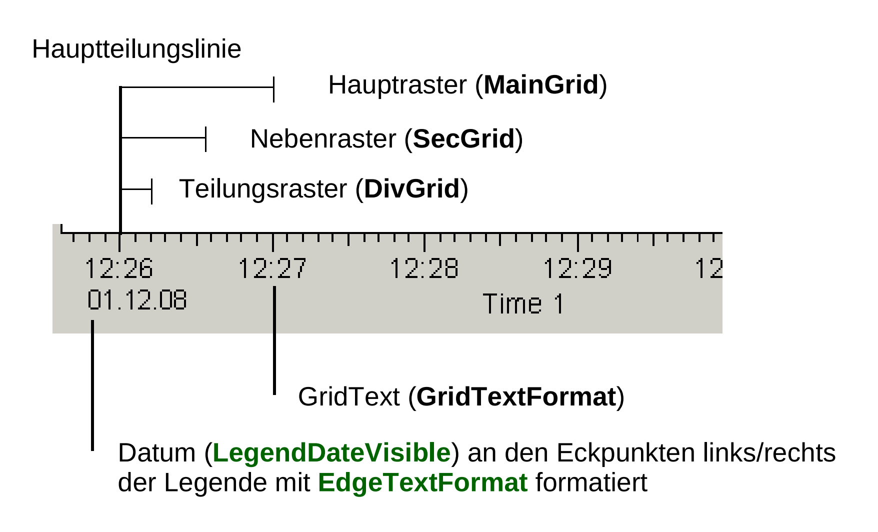
Anmerkungen

Die Angabe des Hauptteilungsraster ist nur relevant, wenn **AutoScaleGrid** auf FALSE gesetzt ist, ansonsten wird das Hauptteilungsraster über die XML-Formatdatei bzw. der Standardeinstellung des *TrendViewers* bestimmt.



Beispiel

In der folgenden Abbildung wird der Zusammenhang zwischen **MainGrid**, **SecGrid** und **DivGrid** sowie **GridTextFormat, EdgeTextFormat** und **Legend-DateVisible** dargestellt:




```
Dim xScale1 As TrendViewer30.CTVXScale
Dim yScale1 As TrendViewer30.CTVYScale
Dim Range1 As TrendViewer30.CTVRange

xScale1 = m_Trend.GetXScale(0)
yScale1 = m_Trend.GetYScale(0)

m_Trend.AutoScaleGrid = False

' define y-Scale
yScale1.MainGrid = 50
yScale1.SecGrid = 2
yScale1.DivGrid = 2

yScale1.GridTextFormat = "%.1f"
yScale1.ToolTipFormat = "%.1f"

' define x-Scale
```



```
Range1 = xScale1.Range

' five minutes before and 1 minutes after
Range1.ActualMinDateTime = Now.AddMinutes(-1)
Range1.ActualMaxDateTime = Now.AddMinutes(1)

' each minute a main-line
xScale1.MainGrid = (60 / 86400)
xScale1.SecGrid = 2
xScale1.DivGrid = 5

xScale1.GridTextFormat = "{0:HH:mm}"
xScale1.ToolTipFormat = "{0:dd.MM.yy HH:mm:ss}"

xScale1.LegendDateVisible = True
xScale1.EdgeTextFormat = "{0:dd.MM.yy}"
```

Siehe auch

SecGrid, DivGrid, GridTextFormat, AutoScaleGrid, GridTextFormat, EdgeTextFormat

Mirrored

`Boolean` Mirrored()

Beschreibung

Spiegelt die x-Skala.

Wertebereich

`TRUE`	Skala wird gespiegelt
`FALSE`	Skala wird in der normalen Orientierung dargestellt

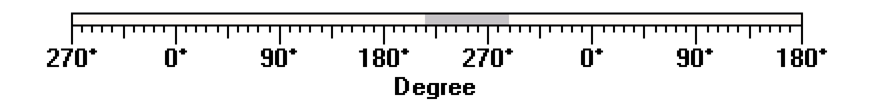
Modulo360Degrees

`Boolean` Modulo360Degrees()

Beschreibung

Die Gradskala wird normalerweise mit beliebigen Grad-Werten dargestellt. Wird diese Eigenschaft auf `TRUE` gesetzt, so werden alle Gradwerte mittels Modulofunktion auf einem Gradbereich zwischen 0° und 360° abgebildet. wie folgendes Beispiel zeigt:





Wertebereich

`TRUE`	Abbildung auf 0° bis 360°
`FALSE`	normale Gradskala

Anmerkung

Diese Einstellung ist nur für Gradskalen relevant.

Siehe auch

Type

Position

`TVPosition` Position()

Beschreibung

Referenz zu einem **TVPosition**-Objekt. Legt fest, wo die x-Skala positioniert wird.

Wertebereich

`siehe` **TVPosition**

Priority

`Integer` Index()

Beschreibung

Anzeigepriorität der x-Skala.

Wertebereich

beliebiger ganzzahliger Wert

Anmerkung

Die Anzeigepriorität kommt besonders in den Randbereichen zum tragen. Hier kann je nach Einstellung des *TrendViewers* zusätzlich auch mit dem **GetMarginFactor (TrendCtrl)** variiert werden, um Überlappungen einzelner Skalen zu vermeiden.

Siehe auch

GetMarginFactor (TrendCtrl)



Range

`TVRange` Range()

Beschreibung

Referenz zu einem **TVRange**-Objekt.Definiert die Grenzen der verschiedenen Skalenbereiche.

Wertebereich

siehe **TVRange**

RangeMoveOnly

`Boolean` RangeMoveOnly()

Beschreibung

Der Darstellungsbereich der x-Skala kann nur mittels Maussteuerung verschoben werden.

Wertebereich

`TRUE`	nur Verschieben zugelassen
`FALSE`	Verschieben, Vergrößern und Verkleinern des Ausschnitts zugelassen, Voreinstellung

Anmerkung

Für **RangeMoveOnly** = `TRUE` haben die Zoomfunktionen des Trend-Fensters keine Auswirkung auf die x-Skala. Der Darstellungsbereich kann über die erweiterten Zoomfunktionen des Trend-Fensters nur durch Bewegen des Trend-Ausschnitts verändert werden.

Siehe auch

Interactive, **Zoom**

Res

`Double` Res()

Beschreibung

Minimale Auflösung, die durch die Skalenteilung dargestellt werden kann.

Wurde die x-Skala als Zeitskala definiert, kann nur die Eigenschaft **RangeRes** geändert werden, die Eigenschaften **Max, Min, MaxDateTime, MinDateTime** haben festgelegte Presetwerte, die durch das interne Zeitformat von Windows festgelegt werden.

Voreinstellungen (Datenskala)

	Datenskala	Zeitskala
Max, Min, MaxDateTime, MinDateTime	200, -200	Presetwert, Presetwert
Res	0,001	0,001



Siehe auch

TVRange

SecGrid

`Integer` SecGrid()

Beschreibung

Das Nebenraster legt Anzahl der Unterteilungen innerhalb des Hauptrasters (**MainGrid**) fest.

Wertebereich

beliebiger ganzzahliger Wert

Anmerkungen

Eine genauere Erläuterung dieser Eigenschaft findet sich innerhalb der Beschreibung von **MainGrid (TVXScale)** .

SecGrid ist nur relevant, wenn AutoScaleGrid = FALSE ist, ansonsten werden die Einstellungen aus der XML-Formatdatei bzw. den Standardeinstellungen des *TrendViewers* übernommen.

Siehe auch

MainGrid, **DivGrid, AutoScaleGrid, MainGrid (TVXScale)**

Shared

`String` Shared()

Beschreibung

Bestimmt, ob alle Kurven dieselben x-Daten benutzen (TRUE) oder ob jede Kurve einen eigenen x-Datensatz benötigt (FALSE)

Wertebereich

`TRUE`	alle Kurven verwenden die gleichen x-Daten
`FALSE`	alle Kurven besitzen eigene x-Daten

Anmerkung

Diese Eigenschaft bestimmt eine fundamentale Eigenschaft der internen Datenhaltung:

Ist **Share** gleich `TRUE`, wird nur ein Satz von x-Daten gespeichert, die Datenpunkte aller aktivierten Kurven bilden dann zusammen einen Datenpunkt der Form (X,Ya,Yb,...,Yn). Für diese Datenhaltung ist es möglich, den Trend als Phasendiagramm (siehe auch Kapitel 2.3.2) darzustellen.

Ist **Share** gleich `FALSE`, wird für jede Kurve ein eigener Datensatz angelegt. Die Kurven sind dann alle von einander unabhängig und besitzen jede unabhängige Datenpunkte der Form (Xa,Ya).



Bei der Arbeit mit den Kurven-Daten muß diese Eigenschaft immer berücksichtigt werden, da bei Fehlbedienung leicht Inkonsistenzen im Datensatz entstehen können. Hierauf wird in den Anmerkungen zu den jeweiligen Eigenschaften und Methoden genauer eingegangen.

Bei Änderung der Eigenschaft IdenticalXData, werden automatisch alle internen Datensätze gelöscht.

Siehe auch

Range, Res, TVData

Text

`String` Text()

Beschreibung

Skalentext zur Beschreibung der x-Skala.

Wertebereich

beliebiger Text

Anmerkung

Wenn der Platz zur Darstellung des Textes nicht ausreicht, wird die Beschriftung abgeschnitten und mit drei Punkten gekennzeichnet.

Siehe auch

LegendDateVisible, EdgeTextFormat, ScaleLegends, TextAlign

TextAlign

`enumTextAlign` TextAlign()

Beschreibung

Textausrichtung des Skalentextes in der Legende (**enumTextAlign**).

Wertebereich

`0-taCenter`	Text zentriert
`1-taLeft`	Text links
`2-taRight`	Text rechts

Siehe auch

Shared, ScaleLegends, enumTextAlign

TimeZoneName

`string` TimeZoneName()

Beschreibung

Legt die Zeitzone für die Skala fest.



Siehe auch

TVData, SystemTimeZones

ToolTipFormat

`string` GridTextFormat()

Beschreibung

Bestimmt das Format der Werteanzeige für den ToolTip (**DataWindows**).

Wertebereich

Formatierungstyp in C-Notation z.B. "%d.%m.%y"

Anmerkung

Eine genauere Erläuterung dieser Eigenschaft findet sich innerhalb der Beschreibung von **MainGrid (TVXScale)**.

Siehe auch

DataWindows, MainGrid (TVXScale)

Type

`enumScaleTypes` Type()

Beschreibung

Mit dieser Eigenschaft kann der Datentyp der x-Skala (Dezimal, Zeit-, Grad- oder Logarithmusdarstellung) ausgewählt werden.

Wertebereich

`0 -stDecimal`	Dezimalskala
`1 -stTime`	Zeitskala
`2 -stDegree`	Gradskala
`3 -stLogarithmic`	Logarithmusskala

Siehe auch

enumScaleTypes, **AbsoluteTime**, **Modulo360Degrees**

Visible

`Boolean` Visible()

Beschreibung

Mit dieser Eigenschaft kann die x-Skala ein- und ausgeblendet werden.

Wertebereich

TRUE	sichtbar
FALSE	nicht sichtbar



Anmerkungen

Diese Eigenschaft kann nur geändert werden, wenn **AutoScaleVisibility** = `FALSE` ist, andernfalls übernimmt die Trend-Komponente die Aktivierung der Skalen selbst.

Es können nur x-Skalen sichtbar geschaltet werden, die zuvor aktiviert (s. **Enabled**) wurden.

Siehe auch

AutoScaleVisibility, **Enabled**, **TVPosition**



5.1.3 TVYScale

Eigenschaften des y-Skalenobjekts :

AuxCount	Anzahl der Hilfslinien
AuxLine	Stil der Hilfslinie (normale Darstellung oder Ausrichtung an der Hauptteilungslinie)
AuxLineStyle	Referenz zu einem **TVLineStyle-**Objekt. (Definiert den Stil einer Hilfslinie)
AuxStart	Startpunkt der ersten Hilfslinie
AuxStep	Schrittweite der Hilfslinien in Prozent
DataRangeEnabled	zeigt (`TRUE`) oder versteckt (`FALSE`) die Anzeige des aktuellen Wertebereichs innerhalb des Übersichtbalkens (nicht änderbar für **AutoScaleVisibility**=`TRUE`)
DivGrid	Definiert das Teilungsraster. Das Teilungsraster legt Anzahl der Unterteilungen innerhalb des Nebenrasters (**SecGrid**) fest
EdgeTextFormat	Format für die Textausgabe links/rechts in der Legende
Enabled	aktiviert (`TRUE`) oder deaktiviert (`FALSE`) die Skala (änderbar nur für **AutoScaleVisibility**=`FALSE`)
ForeColor	Vordergrundfarbe der Skala
GridTextFormat	Textformat der Werteanzeige an den Hauptteilungslinien
Index	Instanz der Skala
Interactive	aktiviert (`TRUE`) oder deaktiviert (`FALSE`) die Mausbedienung
MainGrid	Definiert das Hauptraster. Ein Hauptraster ist der Abstand zwischen zwei Hauptteilungslinien
Mirrored	Spiegelt die Skala
Modulo360Degrees	zeigt bei einer Gradskala nur Werte zwischen 0° und 360° (bildet immer positives Modulo 360°)
Position	Referenz zu einem **TVPosition-**Objekt (Positionierung der Skala)
Priority	Legt die Priorität der Skala fest
Range	Referenz zu einem **TVRange-**Objekt (Datenbereich der Skala)
RangeMoveOnly	gestattet für (`TRUE`) nur die Verschiebung des Wertebereichs
Res	maximale Anzeigegenauigkeit (nicht persistent für Zeitachsen)
SecGrid	Definiert das Nebenraster. Das Nebenraster legt Anzahl der Unterteilungen innerhalb des Hauptrasters (**MainGrid**) fest
Text	Name der Skala (erscheint in der Legende)
TextAlign	Textausrichtung in der Legende
ToolTipFormat	Format für die Text-u.Werteausgabe im Tooltip (Datenfenster)
Type	bestimmt das Anzeigeformat der Skala
Visible	zeigt (`TRUE`) oder versteckt (`FALSE`) die Skala (nur



	für **AutoScaleVisibility**=`FALSE`)
XYScaleEnabled	Skala als XY-Skala aktivieren (`TRUE`)

AuxCount

`Integer` AuxCount()

Beschreibung

Definiert die Anzahl von Hilfslinien, die gezeichnet werden sollen.

Wertebereich

-1	unbegrenzte Anzahl von Hilfslinien (z.B. für kontinuierliche Rasterung)
<=0	keine Hilfslinien
>0	Anzahl gezeichneter Hilfslinien

Anmerkungen

Bei der Erzeugung von y-Skalen-Hilfslinien werden ab der Position **AuxStart** die durch **AuxCount** bestimmte Anzahl von Hilfslinien mit der Schrittweite **AuxStep** gezeichnet.

Eine genauere Erläuterung mit Beispielen dieser Eigenschaft findet sich innerhalb der Beschreibung von **AuxLine (TVXScale)**.

Siehe auch

AuxStart, AuxStep, AuxLine, AuxLineStyle, AuxLine (TVXScale)

AuxLine

`enumAuxLineTyp` AuxLine()

Beschreibung

Definiert die Anordnung der Hilfslinien.

Wertebereich

`0`	Hilfslinien werden über die Anzahl der Linien (**AuxCount**), Start (**AuxStart**) und Step (**AuxStep**) definiert
`1`	Hilfslinien werden über **AuxStep** an den Teilungslinien (**DivGrid, MainGrid, SecGrid**) der Skalen ausgerichtet)

Anmerkungen

Bei AuxLine = 1 hat die Eigenschaft **AuxStart** und **AuxCount** keine Bedeutung !

Eine genauere Erläuterung mit Beispielen dieser Eigenschaft findet sich innerhalb der Beschreibung von **AuxLine (TVXScale)**.

Siehe auch

AuxStart, AuxStep, AuxCount, AuxLineStyle, AuxLine (TVXScale)



AuxLineStyle

`TVLineStyle` AuxLineStyle()

Beschreibung

Referenz auf ein **TVLineStyle**-Objekt. Legt den Linienstil der y-Skalen-Hilfslinien fest.

Wertebereich

siehe **TVLineStyle**

Siehe auch

AuxStart, AuxStep, AuxCount, AuxLine, TVLineStyle

AuxStart

`Double` AuxStart()

Beschreibung

Legt den Startwert für das Zeichnen der Hilfslinien fest.

Wertebereich

beliebiger Wert

Anmerkung

Eine genauere Erläuterung dieser Eigenschaft findet sich innerhalb der Beschreibung von **AuxLine (TVXScale)**.

Siehe auch

AuxStep, AuxCount, AuxLine, AuxLineStyle, AuxLine (TVXScale)

AuxStep

`Double` AuxStep()

Beschreibung

Legt die Schrittweite zwischen den Hilfslinien fest.

Wertebereich

beliebiger Wert

Anmerkung

Eine genauere Erläuterung dieser Eigenschaft findet sich innerhalb der Beschreibung von **AuxLine (TVXScale)**.

Siehe auch

AuxStart, AuxCount, AuxLine, AuxLineStyle, AuxLine (TVXScale)



DataRangeEnabled

`Boolean` DataRangeEnabled()

Beschreibung

Schaltet die Darstellung des aktuellen Datenintervalls im Übersichtsbalken ein. Das Datenintervall wird durch die Eigenschaften **DataMax, DataMin, DataMax-DateTime, DataMinDateTime** festgelegt.

Wertebereich

`FALSE`	nicht eingeschaltet, Voreinstellung
`TRUE`	Datenintervall wird dargestellt

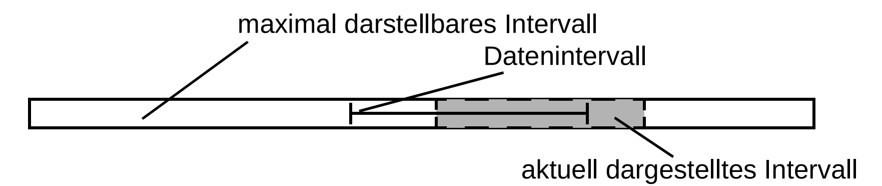
Anmerkungen

Bei der Darstellung von Meßwertkurven ist es hilfreich, den größten und den kleinsten Wert innerhalb der Meßwertmenge zu kennen, damit die Meßwert-kurve mit der besten Auflösung dargestellt werden kann. Außerdem hilft die Darstellung des Datenintervalls festzustellen, welcher Ausschnitt des Datenintervalls im Vergleich zur Skala dargestellt wird.

Der Datenbereich kann nur während der Laufzeit festgelegt werden. Zur Entwicklungszeit wird jedoch ein Standard Datenbereich festgelegt, um das spätere Aussehen der Skala anzudeuten.

Beispiel (Datenskala)

Der Zusammenhang zwischen dem maximal darstellbaren Intervall, dem aktuell dargestellten Intervall und dem Datenintervall geht aus folgender Skizze hervor:



Bei dieser Konstellation zeigt die Skala aktuell nur einen kleinen Ausschnitt aus der gesamten möglichen Grafik an. Von dem Datenintervall ist ebenfalls nur ein Ausschnitt zu sehen.

Programm-Beispiel

```
Dim YScale As TrendViewer30.CTVYScale
Dim YRange As TrendViewer30.CTVRange

YScale = m_Trend.GetYScale(0)
YRange = YScale.Range

YScale.DataRangeEnabled = False
YRange.DataMin = 10
YRange.DataMax = 100
```



```
YScale.DataRangeEnabled = True
```

Siehe auch

TVRange

DivGrid

`Integer` DivGrid()

Beschreibung

Definiert das Teilungsraster. Das Teilungsraster legt Anzahl der Unterteilungen innerhalb des Nebenrasters (**SecGrid**) fest.

Wertebereich

beliebiger ganzzahliger Wert

Anmerkungen

Eine genauere Erläuterung dieser Eigenschaft findet sich innerhalb der Beschreibung von **MainGrid (TVXScale)**.

DivGrid ist nur relevant, wenn AutoScaleGrid gleich FALSE ist, ansonsten werden die Einstellungen aus der XML-Formatdatei bzw. den Standardeinstellungen des *TrendViewers* übernommen.

Siehe auch

MainGrid, SecGrid, AutoScaleGrid, MainGrid (TVXScale)

EdgeTextFormat

`string` EdgeTextFormat()

Beschreibung

Bestimmt das Format der Werteausgabe "links und rechts" in der Legende.

Wertebereich

Formatierungstyp in .Net-Notation z.B. "{0:HH:mm:ss}"

Anmerkungen

EdgeTextfFormat ist nur relevant, wenn **AutoScaleGrid** gleich FALSE ist, ansonsten werden die Einstellungen aus der XML-Formatdatei bzw. den Standardeinstellungen des *TrendViewers* übernommen.

Eine genauere Erläuterung dieser Eigenschaft findet sich innerhalb der Beschreibung von **MainGrid (TVXScale)**.

AutoScaleGrid, LegendDateVisible, MainGrid (TVXScale)



Enabled

`Boolean` Enabled()

Beschreibung

Aktiviert oder deaktiviert eine y-Skala.

Wertebereich

`TRUE`	Skala ist aktiviert
`FALSE`	Skala ist deaktiviert

Anmerkungen

Diese Eigenschaft kann nur geändert werden, wenn **AutoScaleVisibility** = `FALSE` ist, andernfalls übernimmt die Trend-Komponente die Aktivierung der Skalen selbst.

Es können nur y-Skalen sichtbar geschaltet (s.**Visible**) werden, die zuvor aktiviert wurden.

Siehe auch

AutoScaleVisibility, **Visible**

ForeColor

`System.Drawing.Color` ForeColor()

Beschreibung

Skalen und Beschriftungsfarbe der x-Skala.

Wertebereich

wird durch die Windowsfarbpalette vorgegeben

Siehe auch

BackColor

GridTextFormat

`string` GridTextFormat()

Beschreibung

Bestimmt das Format der Werteanzeige für die Hauptraster.

Wertebereich

Formatierungstyp in C-Notation z.B. "{0:HH:mm:ss}"

Anmerkungen

Eine genauere Erläuterung dieser Eigenschaft findet sich innerhalb der Beschreibung von **MainGrid (TVXScale)**.



GridTextFormat ist nur relevant, wenn AutoScaleGrid gleich FALSE ist, ansonsten werden die Einstellungen aus der XML-Formatdatei bzw. den Standardeinstellungen des *TrendViewers* übernommen.

MainGrid (TVXScale), AutoScaleGrid

Index

`Short` Index()

Beschreibung

Liefert den Indexwert der y-Skala.

Wertebereich

`0..MaxYScales`

Interactive

`Boolean` Interactive()

Beschreibung

Schaltet die interaktive Veränderung des Darstellungsbereichs der y-Skala mit Hilfe der Maus ein (`TRUE`) bzw. aus (`FALSE`).

Wertebereich

`TRUE`	Benutzer kann den Darstellungsbereich der y-Skala mit Hilfe der Maus verändern
`FALSE`	Änderung gesperrt

Anmerkung

Die Zoomfunktionen des Trend-Fensters haben bei **Interactive** = `FALSE` keine Auswirkung auf die y-Skala.

Siehe auch

Zoom (TrendCtrl)

MainGrid

`Double` DivGrid()

Beschreibung

Das Hauptraster definiert den Abstand zwischen zwei Hauptteilungslinien.

Wertebereich

beliebiger Wert



Anmerkungen

Die Angabe des Hauptteilungsraster ist nur relevant, wenn **AutoScaleGrid** auf FALSE gesetzt ist, ansonsten wird das Hauptteilungsraster über die XML-Formatdatei bzw. der Standardeinstellung des *TrendViewers* bestimmt.

Eine genauere Erläuterung dieser Eigenschaft findet sich innerhalb der Beschreibung von **MainGrid (TVXScale)**.

Siehe auch

SecGrid, DivGrid, GridTextFormat, AutoScaleGrid

Mirrored

`Boolean` Mirrored()

Beschreibung

Spiegelt die y-Skala.

Wertebereich

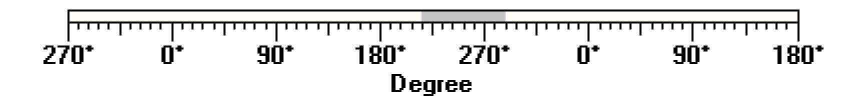
`TRUE`	Skala wird gespiegelt
`FALSE`	Skala wird in der normalen Orientierung dargestellt

Modulo360Degrees

`Boolean` Modulo360Degrees()

Beschreibung

Die Gradskala wird normalerweise mit beliebigen Grad-Werten dargestellt. Wird diese Eigenschaft auf `TRUE` gesetzt, so werden alle Gradwerte mittels Modulofunktion auf einem Gradbereich zwischen 0° und 360° abgebildet. wie folgendes Beispiel zeigt:



Wertebereich

`TRUE`	Abbildung auf 0° bis 360°
`FALSE`	normale Gradskala

Anmerkung

Diese Einstellung ist nur für Gradskalen relevant.

Siehe auch

Type



Position

TVPosition Position()

Beschreibung

Referenz zu einem **TVPosition**-Objekt. Legt fest, wo die x-Skala positioniert wird.

Wertebereich

siehe **TVPosition**

Priority

Integer Index()

Beschreibung

Anzeigepriorität der y-Skala.

Wertebereich

beliebiger ganzzahliger Wert

Anmerkung

Die Anzeigepriorität kommt besonders in den Randbereichen zum tragen. Hier kann je nach Einstellung des *TrendViewers* zusätzlich auch mit dem **GetMargin-Factor** variiert werden, um Überlappungen einzelner Skalen zu vermeiden.

Siehe auch

GetMarginFactor

Range

TVRange Range()

Beschreibung

Referenz zu einem **TVRange**-Objekt.Definiert die Grenzen der verschiedenen Skalenbereiche.

Wertebereich

s. **TVRange**

RangeMoveOnly

Boolean YScMoveOnly()

Beschreibung

Der Darstellungsbereich der y-Skala kann nur mittels Maussteuerung verschoben werden.



Wertebereich

`TRUE`	nur Verschieben zugelassen
`FALSE`	Verschieben, Vergrößern und Verkleinern des Ausschnitts zugelassen, Voreinstellung

Anmerkung

Für **RangeMoveOnly** = `TRUE` haben die Zoomfunktionen des Trend-Fensters keine Auswirkung auf die y-Skala. Der Darstellungsbereich kann über die erweiterten Zoomfunktionen des Trend-Fensters nur durch Bewegen des Trend-Ausschnitts verändert werden.

Siehe auch

Interactive, **Zoom**

Res

`Double` Res()

Beschreibung

Minimale Auflösung, die durch die Skalenteilung dargestellt werden kann.

Voreinstellungen (Datenskala)

Max, Min, MaxDateTime, MinDateTime 200, -200
Res 0,001

Wertebereich

beliebiger Wert

Siehe auch

TVRange

SecGrid

`Integer` SecGrid()

Beschreibung

Das Nebenraster legt Anzahl der Unterteilungen innerhalb des Hauptrasters (**MainGrid**) fest.

Wertebereich

beliebiger ganzzahliger Wert

Anmerkungen

Eine genauere Erläuterung dieser Eigenschaft findet sich innerhalb der Beschreibung von **MainGrid (TVXScale)**.

SecGrid ist nur relevant, wenn AutoScaleGrid gleich FALSE ist, ansonsten werden die Einstellungen aus der XML-Formatdatei bzw. den Standardeinstellungen des *TrendViewers* übernommen.



Siehe auch

MainGrid, **DivGrid, AutoScaleGrid, MainGrid (TVXScale)**

Text

`string` YScText()

Beschreibung

Skalentext zur Beschreibung der y-Skala.

Wertebereich

beliebiger Text

Anmerkung

Wenn der Platz zur Darstellung des Textes nicht ausreicht, wird die Beschriftung abgeschnitten und mit drei Punkten gekennzeichnet.

Siehe auch

ScaleLegends, TextAlign

TextAlign

`enumTextAlign` TextAlign()

Beschreibung

Textausrichtung des Skalentextes in der Legende.

Wertebereich

`0-taCenter`	Text zentriert
`1-taLeft`	Text links
`2-taRight`	Text rechts

Siehe auch

Text, ScaleLegends, enumTextAlign

ToolTipFormat

`string` GridTextFormat()

Beschreibung

Bestimmt das Format der Werteanzeige für den ToolTip (**DataWindows**).

Wertebereich

Formatierungstyp in C-Notation z.B. "{0:0.00}"

Anmerkung

Eine genauere Erläuterung dieser Eigenschaft findet sich innerhalb der Beschreibung von **MainGrid (TVXScale)**.



Siehe auch

DataWindows

Type

`enumScaleTypes` YScScaleType()

Beschreibung

Mit dieser Eigenschaft kann der Datentyp der y-Skala (Winkel-, Dezimal oder Logarithmusdarstellung) ausgewählt werden.

Wertebereich

`0 -stDecimal`	Dezimalskala
`2 -stDegree`	Gradskala
`3 -stLogarithmic`	Logarithmusskala

Anmerkung

Die Zeitdarstellung ist für die y-Skala gesperrt.

Siehe auch

enumScaleTypes, **Modulo360Degrees**

Visible

`Boolean` YScVisible()

Beschreibung

Mit dieser Eigenschaft kann die y-Skala ein- oder ausgeblendet werden.

Wertebereich

`TRUE`	sichtbar
`FALSE`	nicht sichtbar

Anmerkungen

Diese Eigenschaft kann nur geändert werden, wenn **AutoScaleVisibility** = `FALSE` ist, andernfalls übernimmt die Trend-Komponente die Aktivierung der Skalen selbst.

Es können nur y-Skalen sichtbar geschaltet werden, die zuvor aktiviert (s. **Enabled**) wurden.

Siehe auch

AutoScaleVisibility, **Enabled**, **TVPosition**



XYScaleEnabled

`Boolean` XYScaleEnabled()

Beschreibung

Legt fest, ob die jeweilige y-Skala als x-Basis für ein Phasendiagramm verwendet werden kann, also direkt über der x-Skala platziert werden darf.

Wertebereich

TRUE	die y-Skala darf als x-Basis verwendet werden.
FALSE	die y-Skala darf nicht als x-Basis verwendet werden.

Anmerkungen

Dieses Attribut ist nur wirksam, wenn **Shared** = TRUE ist.

Siehe auch

Shared, TVPosition



5.1.4 TVRange

Eigenschaften des Range-Objekts :

Max, Min, MaxDateTime, MinDateTime
maximal, minimal möglicher Skalenwert
(nicht persistent für Zeitachsen)

ActualMax, ActualMin, ActualMaxDateTime, ActualMinDateTime
Maximum, Minimum des aktuell sichtbaren Skalenbereichs

DataMax, DataMin, DataMaxDateTime, DataMinDateTime
Maximum, Minimum des aktuellen Wertebereichs der Kurven, die der Skala zugeordnet sind

---

Max, Min, MaxDateTime, MinDateTime

`Double` Max()

`Double` Min()

`DateTime` MaxDateTime()

`DateTime` MinDateTime()

Beschreibung

Größter bzw. kleinster Wert, den der Skalenendwert bzw. der Skalenanfangswert annehmen kann.

Wertebereich

Es können nur Werte außerhalb des durch **ActualMax, ActualMin, ActualMaxDateTime, ActualMinDateTime** definierten Darstellungsbereichs zugewiesen werden.

Voreinstellungen

	Datenskala	Zeitskala
Max, Min	200, -200	Presetwert, Presetwert

Anmerkungen

Mit diesen Eigenschaften werden die Grenzen der Skalendarstellung festgelegt. Der Benutzer kann innerhalb der Applikation nur Einstellungen vornehmen, die innerhalb der vorgegebenen Grenzen liegen.

Zur Änderung der Wertebereiche ist die in der Anmerkung zu **ActualMax, ActualMin, ActualMaxDateTime, ActualMinDateTime** beschriebene Vorgehensweise zu beachten.

Siehe auch

ActualMax, ActualMin, ActualMaxDateTime, ActualMinDateTime



ActualMax, ActualMin, ActualMaxDateTime, ActualMinDateTime

`Double` Max()

`Double` Min()

`DateTime` Max()

`DateTime` Min()

Beschreibung

Grenzen des aktuell dargestellten Skalenbereichs. Je nach Skalentyp wird ein Wert oder ein Zeitpunkt eingetragen.

Wertebereich

Es können nur Werte innerhalb des durch **Max, Min, MaxDateTime, MinDateTime** definierten maximalen Darstellungsbereichs zugewiesen werden.

Anmerkungen

Alle Wertebereiche der Skala werden vor jedem Zeichenvorgang der Komponente auf ihre Konsistenz überprüft und gegebenenfalls verworfen.

Sollte z.B. der übergebene max Wert größer als der momentan min Wert sein, erfolgt keine Anpassung des max Wertes. Der Anwender muss dies daher zuvor überprüfen, und gegebenenfalls die Reihenfolge der zu übermittelnde Werte umkehren, d.h. zuerst den min Wert, dann den max Wert anpassen.

Programm-Beispiel

```
Dim XScale1 As TrendViewer30.CTVXScale
Dim YScale1 As TrendViewer30.CTVYScale
Dim XRange As TrendViewer30.CTVRange
Dim YRange As TrendViewer30.CTVRange

XScale1 = m_Trend.GetXScale(0)
XRange = XScale1.Range

YScale1 = m_Trend.GetYScale(0)
YRange = YScale1.Range

XRange.ActualMinDateTime = Now.AddHours(-2.0)
XRange.ActualMaxDateTime = Now.AddHours(2.0)

YRange.ActualMin = 10
```



```
YRange.ActualMax = 100
```

Siehe auch

Max, Min, MaxDateTime, MinDateTime, **DataMax, DataMin, DataMaxDateTime, DataMinDateTime**

DataMax, DataMin, DataMaxDateTime, DataMinDateTime

`Double` DataMax()

`Double` DataMin()

`DateTime` DataMinDateTime()

`DateTime` DataMinDateTime()

Beschreibung

Größter bzw. kleinster Wert des Datenintervalls.

Wertebereich

Min	minimaler Wert
Max	maximaler Wert

Anmerkungen

Wie bereits in den Anmerkungen zu **ActualMax, ActualMin, ActualMaxDateTime, ActualMinDateTime** beschrieben, werden die Wertebereiche bei jedem Zeichenvorgang der Komponente überprüft. Der Datenbereich wird jedoch nur überprüft, wenn er durch **DataRangeEnabled (TVXScale)** bzw. **DataRangeEnabled (TVYScale)** bereits aktiviert wurde.

Programm-Beispiel

siehe **DataRangeEnabled (TVXScale)** bzw. **DataRangeEnabled (TVYScale)**

Siehe auch

Max, Min, MaxDateTime, MinDateTime, **ActualMax, ActualMin, ActualMaxDateTime, ActualMinDateTime**



5.1.5 TVPosition

Eigenschaften des Positionierung-Objekts:

Col	legt die Spalte fest, in der die Skala positioniert werden soll
Group	legt die Gruppe fest, in welche eine Skala plaziert werden soll
Row	legt die Zeile fest, in der die Skala positioniert werden soll
Size	legt die prozentuale Größe der Skala fest.

Col

`Short` Col()

Beschreibung

Legt die Spalte fest, in der die Skala angezeigt wird.

Wertebereich

`0-31`	1. - 32. Spalte

Anmerkung

Eine genauere Erläuterung dieser Eigenschaft findet sich innerhalb der Beschreibung der Methode **SetPosition**.

Siehe auch

SetPosition, Group, Row, Size

Group

`Short` Group()

Beschreibung

Legt die Gruppe fest, in der die Skala angezeigt wird.

Wertebereich

`0-sgTop`	Skala liegt an der Trendoberseite
`1-sgBottom`	Skala liegt an der Trendunterseite
`2-sgLeft`	Skala liegt links
`3-sgRight`	Skala liegt rechts

Anmerkung

Eine genauere Erläuterung dieser Eigenschaft findet sich innerhalb der Beschreibung der Methode **SetPosition**.



Siehe auch

SetPosition, Col, Row, Size

Row

`Short` Col()

Beschreibung

Legt die Zeile fest, in der die Skala angezeigt wird.

Wertebereich

`0-7`	1.- 8. Zeile

Anmerkung

Eine genauere Erläuterung dieser Eigenschaft findet sich innerhalb der Beschreibung der Methode **SetPosition**.

Siehe auch

SetPosition, Group, Col, Size

Size

`Double` Col()

Beschreibung

Legt die Größe der Skala in Prozent fest.

Wertebereich

`0-`	Skala nicht sichtbar
`>0..100`	prozentuale Größe einer Skala

Anmerkung

Eine genauere Erläuterung dieser Eigenschaft findet sich innerhalb der Beschreibung der Methode **SetPosition**.



Siehe auch

SetPosition, Group, Col, Row

5.1.6 TVCurve

Eigenschaften des Kurven-Objekts :

AuxLineStyle	Referenz zu einem **TVLineStyle-**Objekt (Definiert den Stil einer Hilfslinie)
Average	aktiviert (`TRUE`) oder deaktiviert (`FALSE`) die Mittelwertsdarstellung für die komprimierte Kurvendarstellung
AverageWithThreshold	aktiviert (`TRUE`) oder deaktiviert (`FALSE`) die Mittelwertsdarstellung für die komprimierte Kurvendarstellung ab einer bestimmten Anzahl von Datenpunkten.
AverageWithThreshold	Legt die Bandbreite für die sichtbare Anzahl an Kurvenpunkten einer Hüllkurve bzw. einer Mittelwertskurve fest
Data	Referenz zu einem **TVData**-Objekt. (beinhaltet alle Kurvenpunkte)
DrawSymbolAlways	Zeichnet ein Symbol immer unabhängig des Statuswertes
DrawSymbolWithCurveColor	Ein Symbol wird ausschließlich in der Kurvenfarbe gezeichnet
Enabled	aktiviert (`TRUE`) oder deaktiviert (`FALSE`) die Kurve
Envelope	Hüllkurve (`TRUE`) einschalten
EnvelopeThreshold	Schwellwert für die Hüllkurve. (Wert, ab welchem eine komprimierte Kurvendarstellung angezeigt wird)
FillStyle	Referenz zu einem **TVFillStyle**-Objekt. (Definiert den FüllStil der Kurve)
FloodCurve	Referenz zu einer **TVCurve** (referenziert die Kurve, zu welcher geflutet werden soll)
FloodStyle	Flutstil
FloodYValue	Wert (Gerade auf der y-Skala), gegen den geflutet werden soll
Index	Instanz der Kurve
LineStyle	Referenz zu einem **TVLineStyle-**Objekt. (Definiert den Stil der Kurve)
Moveable	Kurve/Kurvenpunkte können bewegt (`TRUE`) werden.
Priority	Priorität der Kurve
Ruler	Referenz zu einem **TVRuler**-Objekt (Kurvenlineal)
RulerViewSync	Die Sichtbarkeit eines Kurvenlineals wird an die Sichtbarkeit der Kurve gekoppelt
SetStateStyle	Referenz zu einem **TVStateStyle**-Objekt (Status eines Kurvenpunktes)
StateFromPoint	Darstellung der Statuswerte
StateStyleCount	Statusanzahl für die Kurvenpunkte (< 512)
Symbol	Nummer des Kurvensymbols (**enumCurveSymbols**)
SymbolStep	Abstand zwischen zwei Kurvensymbolen
Text	Name der Kurve



TypeTrendCurve	legt den Typ einer Kurve fest (z.B. linear, binär etc.)
Visible	zeigt (`TRUE`) oder versteckt (`FALSE`) die Kurve
XScale	Referenz zu einem **TVXScale**-Objekt (Zuordnung der Kurve zu einer x-Skala)
YScale	Referenz zu einem **TVYScale**-Objekt (Zuordnung der Kurve zu einer y-Skala)

AuxLineStyle

`TVLineStyle` AuxLineStyle()

Beschreibung

Referenz auf ein **TVLineStyle**-Objekt. Legt den Linienstil der Kurven-Hilfslinien fest.

Wertebereich

siehe **TVLineStyle**

Siehe auch

GetAuxVisible

Average

`Boolean` Average()

Beschreibung

Legt fest, ob anstatt der normalen Kurve eine Mittelwertkurve angezeigt werden soll.

Wertebereich

`TRUE`	Mittelwertkurve anzeigen
`FALSE`	Mittelwertkurve wird nicht angezeigt

Anmerkungen

Für den Kurventyp "Binär" (s.**TypeTrendCurve**) ist die Eigenschaft **Envelope** deaktiviert.

Siehe auch

EnvelopeThreshold, Envelope, AverageWithThreshold, AverageWith-Threshold

---

AverageWithThreshold

`Boolean` AverageWithThreshold()

Beschreibung

Legt fest, ob ab einer bestimmten Menge von angezeigten Datenpunkten statt der normalen Kurve eine Mittelwertkurve angezeigt werden soll.



Wertebereich

`TRUE`	Mittelwertkurve ab einer bestimmten Datenmenge anzeigen
`FALSE`	Mittelwertkurve wird nicht angezeigt

Anmerkungen

Ab einer bestimmten Anzahl an Datenpunkten einer Kurve sind Kurvendetails durch die beschränkte Auflösung des Ausgabegeräts nicht mehr sichtbar, da Datenpunkte mit z.B. unterschiedlichen x-Werten auf gleiche x-Koordinaten im Ausgabekontext abgebildet werden. In diesem Fall ist es nicht mehr sinnvoll, jeden einzelnen Punkt der Kurve zu zeichnen.

Um bei großen Datenmengen dennoch sinnvolle Graphen zu erhalten, unterstützt die Trend-Komponente zwei Arten zur komprimierten Darstellung von Kurven:

1) Mittelwertkurve:
Die angezeigte Datenmenge wird in Abschnitte unterteilt. Zu jedem dieser Abschnitt werden jeweils die Mittelwerte der x- als auch der y-Daten gebildet und diese übereinander aufgetragen.
2) Hüllkurve:
Die angezeigte Datenmenge wird in Abschnitte unterteilt. Zu jedem dieser Abschnitte wird für die y-Daten der Maximal- und der Minimalwert ermittelt. Die Maximal- und Minimalwerte werden als Hüllkurve über den x-Mittelwerten aufgetragen. Der dadurch festgelegte Bereich kann mit einem Füllmuster/farbe (**FillStyle**) markiert werden.

Das Datenmodell der Trend-Komponente erlaubt die Berechnung der benötigten Minimal-, Maximal- und Mittelwerte ohne nennenswerten Zeitaufwand.

Die komprimierte Darstellung der Kurvendaten ist aktiviert, sobald eine der beiden Komprimierungsmethoden aktiviert wurde. Beide Komprimierungsmethoden können aber auch gleichzeitig für eine Kurve angezeigt werden.

Werden mit der Trend-Komponente große Datenmengen bearbeitet, sollte auf jeden Fall eine Kurvenkomprimierung gewählt werden, da sonst das Zeichnen der Komponente viel Zeit in Anspruch nehmen würde. Die Anzahl der Kurvenpunkte, ab der eine Kurvenkomprimierung eingeschaltet werden soll, kann mit der Eigenschaft **EnvelopeThreshold** eingestellt werden.

Um die interaktive Bedienung der Trend-Komponente voll nutzen zu können, sollte die benötigte Zeit zum Zeichnen des Trends möglichst weit unter einer Sekunde liegen.

Für den Kurventyp "Binär" (s.**TypeTrendCurve**) ist die Eigenschaft **Envelope** deaktiviert.

Siehe auch

EnvelopeThreshold, Envelope, AverageWithThreshold, Average



BandWidth

`Integer` BandWidth()

Beschreibung

Legt die Bandbreite für die sichtbare Anzahl an Kurvenpunkten einer Hüllkurve bzw. einer Mittelwertskurve fest.

Wertebereich

beliebiger ganzzahliger Wert

Siehe auch

Average, Envelope, EnvelopeThreshold

Data

`TVData` Data()

Beschreibung

Referenz zu einem **TVData**-Objekt. Beinhaltet sämtliche Kurven- u. Statuswerte.

Wertebereich

`s.`**TVData**

Anmerkung

Eine genauere Erläuterung dieser Eigenschaft findet sich innerhalb der Beschreibung von **TVData**.

Siehe auch

XGain, XOffset, XValue, XValueLong, XValueDateTime, YGain, YOffset, YValue, YValueLong, State

DrawSymbolAlways

`Boolean` DrawSymbolAlways()

Beschreibung

Aktiviert bzw. deaktiviert das Zeichnen eines Symbols in Abhängigkeit des Statuswertes eines Kurvenpunktes.

Wertebereich

`TRUE`	Zeichnet ein Symbol unabhängig des Statuswertes
`FALSE`	Zeichnet ein Symbol abhängig des Statuswertes

Anmerkungen

Werden keine Statuswerte in einer Kurve verwendet, so hat diese Funktionalität keine direkte Auswirkung auf das Zeichnen von Symbolen sondern wird dann ausschließlich über die Funktion **SymbolStep** gesteuert.



Siehe auch
Symbol,**SymbolStep**,**TVLineStyle**, **TVStateStyle**

DrawSymbolWithCurveColor

`Boolean` DrawSymbolWithCurveColor()

Beschreibung

Aktiviert bzw. deaktiviert das Zeichnen der Farbe eines Symbols in Abhängigkeit des Statuswertes eines Kurvenpunktes.

Wertebereich

`TRUE`	Zeichnet die Farbe eines Symbols unabhängig des Statuswertes
`FALSE`	Zeichnet die Farbe eines Symbols abhängig des Statuswertes

Anmerkungen

Werden keine Statuswerte in einer Kurve verwendet, so hat diese Funktionalität keine direkte Auswirkung auf das Zeichnen der Farbe von Symbolen sondern wird dann ausschließlich über die Farbe der Kurve gesteuert.

Siehe auch
Symbol, **SymbolStep**,**TVLineStyle**, **TVStateStyle**

Enabled

`Boolean` CurveEnabled()

Beschreibung

Aktiviert bzw. deaktiviert die Kurve.

Wertebereich

`TRUE`	Kurve ist aktiviert
`FALSE`	Kurve ist nicht aktiviert

Anmerkungen

Nur eine aktive Kurve kann durch Setzen von **Visible** auf `TRUE` sichtbar werden. Sind die Kurvenknöpfe sichtbar, werden nur Kurvenknöpfe für aktive Kurven angezeigt.

Besonders sorgfältig sollte mit der Aktivierung von Kurven verfahren werden, wenn **Shared** (TVXScale) auf True gesetzt wurde und die Kurvendaten über das Attribute **Data** an die Trend-Komponente übergeben werden. In diesem Fall werden nur Kurvenpunkte übernommen, wenn allen aktiven Kurven ein y-Wert bzw ein Fehlerwert zugewiesen wurde.

Ist die Trend-Eigenschaft **AutoScaleVisibility** gleich `TRUE`, wird mit der Aktivierung der Kurven die Aktivierung der y-Skalen mitgesteuert.

Siehe auch

Shared, **Data**



Envelope

`Boolean` Envelope()

Beschreibung

Legt fest, ob ab einer bestimmten Menge von angezeigten Datenpunkten statt der normalen Kurve eine Hüllkurve angezeigt werden soll.

Wertebereich

`TRUE`	Hüllkurve ab einer bestimmten Datenmenge (**EnvelopeThreshold**) anzeigen
`FALSE`	Hüllkurve wird nicht angezeigt

Anmerkungen

Eine genauere Erläuterung dieser Eigenschaft findet sich innerhalb der Beschreibung von **Average**.

Um eine Hüllkurve in den Hintergrund zu zeichnen um eine mögliche Überdeckung, z.B. mit dem Füllstil, zu vermeiden, kann mit **Priority** zusätzlich die Zeichenpriorität der Kurve festgelegt werden. Für den Kurventyp "Binär" (s. **TypeTrendCurve**) ist die Eigenschaft **Envelope** deaktiviert.

Siehe auch

Average, **EnvelopeThreshold, Priority**

EnvelopeThreshold

`Integer` EnvelopeThreshold()

Beschreibung

Legt den Schwellwert fest, ab dem statt der normalen Kurve eine Hüllkurve angezeigt werden soll.

Wertebereich

beliebiger ganzzahliger Wert
Standardwert : 4000

Anmerkung

Eine genauere Erläuterung dieser Eigenschaft findet sich innerhalb der Beschreibung von **Average**.

Siehe auch

Average, **EnvelopeThreshold**, **Envelope, AverageWithThreshold**



FillStyle

`TVFillStyle` FillStyle()

Beschreibung

Definiert den Füllstil zur Darstellung der jeweiligen Flut (**TVFillStyle**) bzw. Hüllkurve (**Envelope)**.

Wertebereich

siehe **TVFillStyle**

Anmerkung

Eine genauere Erläuterung dieser Eigenschaft findet sich innerhalb der Beschreibung von **TVFillStyle**.

Siehe auch

Envelope

FloodCurve

`TVCurve` FloodCurve()

Beschreibung

Referenz zu einem (anderen) **TVCurve**-Objekt. Das Referenzobjekt entspricht der Kurve, gegen die geflutet werden soll.

Wertebereich

siehe **TVCurve**

Anmerkung

Enthalten die Kurven eine große Anzahl von Werten, kann es bei eingeschaltetem Fluten und je nach Flutstil zu einer verlangsamten Anzeige kommen, insbesondere beim Zoomen oder verschieben einer Kurve.

Siehe auch

FloodYValue, FloodStyle, TVFillStyle

FloodStyle

`enumFloodStyle` FloodStyle()

Beschreibung

Schaltet das Fluten einer Kurve ein/aus und definiert den Flutstil.

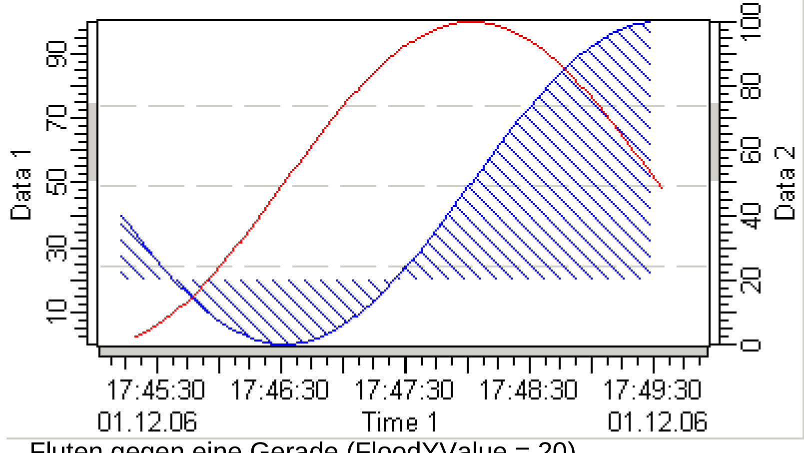
Wertebereich

`0-`	=	Fluten ausgeschaltet
`1-`	=	Fluten gegen eine Gerade
`2-`	=	Fluten gegen eine Kurve

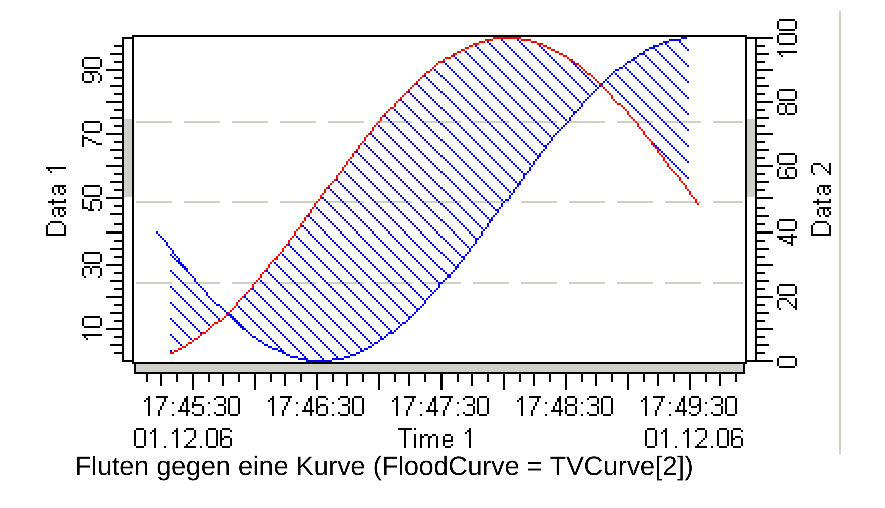


Anmerkungen

Die Voreinstellung ist "0", d.h. Fluten ist ausgeschaltet. Die Eigenschaft **Priority** ermöglicht es, Flutkurven in den Vorder- bzw. Hintergrund zu zeichnen, um ein verdecken anderer Kurven zu vermeiden.



Fluten gegen eine Gerade (FloodYValue = 20)



Fluten gegen eine Kurve (FloodCurve = TVCurve[2])

Siehe auch

FloodCurve, FloodYValue, TVFillStyle, Priority, enumFloodStyle

---

FloodYValue

`Double` FloodYValue()

Beschreibung

Definiert den Y-Wert einer Gerade, gegen die geflutet werden soll.

Wertebereich

`beliebiger Wert`



Anmerkung

Ist der Wert größer/kleiner als die Bereichsgrenze der zugeordneten y-Skala, wird bis zu deren Bereichsgrenze geflutet.

Siehe auch

FloodCurve, FloodStyle, TVFillStyle

Index

`Short` Index()

Beschreibung

Liefert den Index (Kurvennummer) zurück.

Wertebereich

`0..MaxCurves`

LineStyle

`TVLineStyle` LineStyle()

Beschreibung

Referenz zu einem **TVLineStyle**-Objekt. Definiert den Linienstil der Kurve.

Wertebereich

siehe **TVLineStyle**

Moveable

`Boolean` Moveable()

Beschreibung

Legt fest, ob die Kurve bzw. ein einzelner Kurvenpunkt interaktiv über Maussteuerung bewegt werden kann.

Wertebereich

`TRUE`	Kurve/Kurvenpunkt kann bewegt werden.
`FALSE`	Kurve/Kurvenpunkt kann nicht bewegt werden.

Anmerkungen

Je nach dem, ob der Mauszeiger über einem Kurvenpunkt bzw. zwischen zwei Kurvenpunkten steht, erscheint neben dem Mauszeiger ein entsprechendes Symbol, welches anzeigt, ob ein Punkt oder die ganze Kurve mit der rechten Maustaste verschoben werden kann. (siehe Bild unten)

Diese Eigenschaft und Funktionalität ist je nach Lizenzmodell eingeschränkt bzw. nicht verfügbar !



Priority

`Short` Priority()

Beschreibung

Legt die Priorität zum Zeichnen der Kurve fest.

Wertebereich

`0..MaxCurve`

Anmerkungen

Diese Eigenschaft ermöglicht es, gezielt Kurven in den Vorder- bzw. Hintergrund zu zeichnen. Somit kann z.B. beim Fluten oder bei einer Hüllkurve vermieden werden, das andere Kurven mit dem Füllmuster überdeckt werden.

Diese Eigenschaft und Funktionalität ist je nach Lizenzmodell eingeschränkt bzw. nicht verfügbar !

Siehe auch

FloodStyle, Average, Envelope

Ruler

`TVRuler` Ruler()

Beschreibung

Referenz zu einem **TVRuler**-Objekt. Jeder Kurve kann ein separates Lineal zugeordnet werden.

Wertebereich

`siehe` **TVRuler**

RulerViewSync

`Boolean` RulerViewSync()

Beschreibung

Die Sichtbarkeit eines Kurvenlineals wird an die Sichbarkeit der Kurve gekoppelt - dies ermöglicht wahlweise eine getrennte oder gemeinsame Steuerung der Anzeige des Lineals mit der Kurve.

Wertebereich

`TRUE`	Sichtbarkeit des Lineals mit Kurve gekoppelt.
`FALSE`	Sichtbarkeit des Lineals nicht mit Kurve gekoppelt.



StateFromPoint

`Boolean` StateFromPoint()

Beschreibung

Legt fest, wie die Statuswerte dargestellt werden sollen.

Wertebereich

`TRUE`	Status wird ab dem Kurvenpunkt gezeichnet
`FALSE`	Statuswert des nachfolgenden Kurvenwertes wird ab dem aktuellen Kurvenpunkt gezeichnet

Siehe auch

SetStateStyle, TVStateStyle, State (TVData), AddYData

StateStyleCount

`Integer` StateStyleCount()

Beschreibung

Legt die Anzahl der Statuswerte für die Kurve fest.

Wertebereich

`0..512`

Anmerkung

Eine genauere Erläuterung dieser Eigenschaft findet sich innerhalb der Beschreibung von **AddYData**.

Siehe auch

SetStateStyle, TVStateStyle, State (TVData), AddYData

Symbol

`enumCurveSymbols` CurveSymbol()

Beschreibung

Definiert das Kurvensymbol der jeweiligen Kurve.

Wertebereich

siehe **enumCurveSymbols**

Siehe auch

enumCurveSymbols, SymbolStep, TVSymbol



SymbolStep

`Integer` CurveSymbolStep()

Beschreibung

Bestimmt den Abstand zwischen zwei Kurvensymbolen.

Wertebereich

`0`	es werden keine Symbole gezeichnet
`>0`	Schrittweite in Kurvenpunkten zwischen zwei Symbolen

Siehe auch

Symbol

Text

`string` CurveText()

Beschreibung

Beschreibung für die jeweilige Kurve.

Wertebereich

beliebiger Text

Anmerkungen

Die Kurvenbeschreibung sollte möglichst kurz gehalten werden, da ansonsten die Darstellung sowie der Ausdruck der Legende(s. **GetXScale**) zuviel Platz in Anspruch nehmen könnte. Die Kurvenbeschreibung wird im Gegensatz zur Skalenbeschreibung nicht automatisch gekürzt.

Siehe auch

DataWindows, **GetXScale**

TypeTrendCurve

`enumCurveTypes` TypeTrendCurve()

Beschreibung

Legt den Typ einer Kurve fest. (Linear, Binär, Treppenstufen).

Wertebereich

siehe **enumCurveTypes**



Anmerkungen

Wird eine Treppendarstellung ausgewählt, so werden auch Wertebereiche und Mittelwertskurven mit Treppenstufen dargestellt.

Sonderfälle für Binärkurvendarstellung

Wird für eine Kurve die Binärdarstellung ausgewählt, so gilt für die zugeordnete y-Skala und allen anderen Kurven, die der selben y-Skala zugeordnet sind, folgende Definition:

- Alle Kurven auf der selben y-Skala werden automatisch zu Binärkurven
- Hüll- und Mittelwertkurve werden deaktiviert
- Kurvenhilfslinien werden deaktiviert
- Grenzbereich der y-Skala wird automatisch auf 0..1 eingestellt

Siehe auch

enumCurveTypes, Average, Envelope, GetAuxVisible

Visible

`Boolean` CurveVisible()

Beschreibung

Kurve ist sichtbar bzw. versteckt.

Wertebereich

`TRUE`	Kurve ist sichtbar
`FALSE`	Kurve ist versteckt

Anmerkungen

Nur aktive Kurven (s. **Siehe auch Symbol,SymbolStep,TVLineStyle, TVStateStyle**) können sichtbar werden. Ist die Trend-Eigenschaft **AutoScaleVisibility** gleich `TRUE`, wird mit der Sichtbarkeit der Kurven die Sichtbarkeit der Skalen mitgesteuert.

Siehe auch

AutoScaleVisibility, Siehe auch Symbol,SymbolStep,TVLineStyle, TVStateStyle, XScale, YScale

XScale

`TVXScale` XScale()

Beschreibung

Ordnet der Kurve eine der zwei x-Skalen zu.



Wertebereich

TVXScale

Anmerkung

Die Voreinstellung ordnet jeder Kurve die erste x-Skala zu.

Siehe auch

TVXScale, Siehe auch Symbol,SymbolStep,TVLineStyle, TVStateStyle, **Visible, AutoScaleVisibility, YScale**

YScale

`TVYScale` CurveRelatedScale()

Beschreibung

Ordnet der Kurve eine der bis zu max. 512 y-Skalen zu.

Wertebereich

TVYScale

Anmerkung

Ist die Trend-Eigenschaft **AutoScaleVisibility** gleich `TRUE`, wird mit der Sichtbarkeit der Kurven die Sichtbarkeit der y-Skalen mitgesteuert.

Siehe auch

TVYScale, Siehe auch Symbol,SymbolStep,TVLineStyle, TVStateStyle, **Visible, AutoScaleVisibility, XScale**



5.1.7 TVData

Eigenschaften des Kurvenwerte-Objekts:

MaxDataSet	Definiert die maximale Anzahl von Kurvenpunkte, die für eine Kurve im Datensatz gehalten werden
State	Statuswert des Kurvenpunktes
TimeZoneName	Zeitzone der Datensätze
XGain	Gain des x-Wertes
XOffset	Offset des x-Wertes
XValue, XValueLong, XValueDateTime	x-Wert
YGain	Gain des y-Wertes
YOffset	Offset des y-Wertes
YValue, YValueLong	y-Wert

MaxDataSet

`Integer` MaxDataSet()

Beschreibung

Definiert die maximale Anzahl von Kurvenpunkte, die für eine Kurve im Datensatz gehalten werden.

Wertebereich

`0`	die Anzahl der Kurvendaten ist unbegrenzt, in diesem Fall muß die Applikation selbst den Arbeitsspeicher überwachen, damit kein Speicherüberlauf auftritt.
1 und < 2^30-1	die Anzahl maximal gehaltener Kurvenpunkte wird auf diesen Wert begrenzt.

Anmerkungen

Zur Beobachtung längerer Meßreihen ist es notwendig, die gehaltene Datenmenge auf eine maximale Anzahl von Kurvenpunkten zu begrenzen. Erreicht die Anzahl der Kurvenpunkte dieses Limit, werden beim Hinzufügen weiterer Kurvenpunkte die gleiche Anzahl der zugefügten Kurvenpunkten am Anfang des Datensatzes gelöscht, so dass die Gesamtzahl der gehaltenen Kurvenpunkte konstant bleibt.

Ist die x-Skala-Eigenschaft **Shared** gleich `TRUE`, muß der **MaxDataSet** für alle Kurven gleich sein. Daher wird bei jeder Änderung von **MaxDataSet** einer Kurve der übergebene Wert auch für alle anderen Kurven gesetzt.

Ist der neue Wert von **MaxDataSet** kleiner als die Anzahl der bereits existierenden Datensätze, wird eine entsprechende Anzahl Datenpunkte am Anfang des Datensatzes gelöscht.

Siehe auch

Shared, TVData



State

`Integer` State()

Beschreibung

Defniert den Statuswert des Kurvenpunktes. Der Wert, der hier angegeben wird, referenziert die Statuslinie, die in **SetStateStyle** der zugehörigen Kurve definiert wurde.

Wertebereich

beliebiger ganzzahliger Wert

Anmerkungen

Der Status und damit das Zeichnen der Linie ändert sich immer zum Wertepaar hin, d.h.ändert sich der Statuswert im zugehörigen Kurvenpunkt (z.B. "B"), wird die Linie vom vorhergehenden Kurvenpunkt (z.B. "A") mit dem Statuswert des Kurvenpunktes "B" gezeichnet.

Eine genauere Erläuterung dieser Eigenschaft findet sich innerhalb der Beschreibung von **AddYData**.

Siehe auch

SetStateStyle (Kurve), StateFromPoint (Kurve), TVStateStyle, AddYData

TimeZoneName

`string` TimeZoneName()

Beschreibung

Legt die Zeitzone für die aktuellen Datensatz fest.

Wertebereich

exakter Standard oder "daylight" Zeitzonenname

Anmerkungen

Die Zeitzone gilt für den kompletten Datensatz des **TVData**-Objektes. Es ist nicht möglich, für einzelnen Wertepaare unterschiedliche Zeitzonen zu definieren.

Siehe auch

TVData, TVXScale, SystemTimeZones

XGain

`Double` XGain()

Beschreibung

Definiert den Verstärkungsfaktor für die x-Wert einer Kurve.



Wertebereich

beliebiger Wert

Anmerkungen

Der Verstärkungsfaktor wird mit allen x-Werte der zugehörigen Kurve multipliziert. Die Standardeinstellung ist 1.

Eine genauere Erläuterung dieser Eigenschaft findet sich innerhalb der Beschreibung von **YOffset**.

Siehe auch

XValue, XValueLong, XValueDateTime, XOffset, YOffset

XOffset

`Double` XOffset()

Beschreibung

Definiert den Aufsetzpunkt der Kurve für die x-Werte.

Wertebereich

beliebiger Wert

Anmerkungen

Mit dem Festlegen des Aufsetzpunktes kann eine Kurve beliebig in x-Richtung nach links bzw. nach rechts verschoben werden. Die Standardeinstellung ist 0.

Eine genauere Erläuterung dieser Eigenschaft findet sich innerhalb der Beschreibung von **YOffset**.

Siehe auch

XValue, XValueLong, XValueDateTime, XGain, YOffset

XValue, XValueLong, XValueDateTime

`Double` XValue()

`Long` XValueLong()

`DateTime` XValueDateTime()

Beschreibung

Liefert den x-Wert des letzten Datenpunktes bzw. setzt den x-Wert eines neuen Datenpunktes.

Wertebereich

beliebiger Wert



Anmerkungen

Die Kurven-Eigenschaften **State**, **XValue** und **YValue, YValueLong** unterscheiden sich von anderen Properties durch ihre unterschiedliche Verwendung des Lese- und Schreibvorgangs. Dieses Verhalten resultiert aus den Besonderheiten des internen Datenmodells der Trend-Komponente.

Anmerkungen zum internen Datenmodell:

Ein neuer Meßpunkt wird mittels der Eigenschaften **State**, **XValue** und **YValue, YValueLong** sukzessive aufgebaut. Sind alle relevanten Werte des Datenpunkts definiert, wird dieser in einem Zwischenpuffer abgelegt. Die so zwischengespeicherten Daten werden erst beim nächsten Aufbau des Trendbildes in den Hauptspeicher übertragen.

Erzeugung eines neuen Datenpunktes:

Ein Datenpunkt muß vollständig angegeben werden, bevor dieser an den Zwischenspeicher übergeben wird und mit dem Aufbau eines neuen Datenpunktes begonnen werden kann. Wie bereits in den Anmerkungen zur Trend-Eigenschaft **Shared (TVXScale)** ausführlich beschrieben wurde, sind die Werte, aus denen ein Datenpunkt zusammengesetzt wird, von der Trend-Eigenschaft **Shared** abhängig:

Für **Shared** = `TRUE` besteht ein Datenpunkt aus einem einzigen x-Wert und den y- bzw. Statuswerten aller aktivierten Kurven (s. **Siehe auch Symbol,SymbolStep,TVLineStyle, TVStateStyle (TVCurve)**).

Für **Shared** = `FALSE` sind die Datenpunkte der Kurven unabhängig voneinander. Jeder Datenpunkt einer Kurve wird daher durch seinen x-Wert sowie seinen y- bzw. Fehlerwert vollständig definiert.

Vorteil gegenüber den **AddXData**-, **AddYData**-Methoden:

Im Gegensatz zu den **AddXData**-, **AddYData**-Methoden muß der Programmierer nicht selbst für die Konsistenz der Datensätze sorgen. Zwar ist die Logik der hier beschriebenen Methode relativ komplex, jedoch ist bei korrekter Anwendung die Vorgehensweise sehr einfach, wie das anschließende Programm-Beispiel zeigt.

Nachteil gegenüber den **AddXData**-, **AddYData**-Methoden:

Sollen viele Daten gleichzeitig an die Komponente übergeben werden, bremst die OLE-Kommunikation die Übergabe erheblich, in diesem Fall sollten die Daten in Blöcken mittels **AddXData** und **AddYData** übergeben werden.

Das folgende Programm-Beispiel zeigt für beide Zustände von **Shared**, wie Datenpunkte mittels **State**, **XValue** und **YValue, YValueLong** an die Komponente übergeben werden. In diesem Beispiel sind nur die Kurven 1 bis 3 aktiv.

Programm-Beispiel

```
Dim xScale1 As TrendViewer30.CTVXScale

Dim yScale1 As TrendViewer30.CTVYScale

Dim curve As TrendViewer30.CTVCurve

Dim data As TrendViewer30.CTVData
```



```
xScale1 = m_Trend.GetXScale(0)
yScale1 = m_Trend.GetYScale(0)
curve = m_Trend.GetCurve(0)
data = curve.data

'assuming only Curve 1 to Curve 3 are active
If (xScale1.Shared = True) Then
'for shared = TRUE the x-value must be set
data.xValue = newXValue

Dim i As Integer
For i = TrendViewer30.enumCurves.cnCurve1 To
TrendViewer30.enumCurves.cnCurve3
          curve = m_Trend.GetCurve(i)
          data = curve.Data

          data.YValue = newYValue
          data.State = newState
Next i

'data point completed
        Else
'for shared = FALSE the x-value must be _ set
'for every curve

Dim i As Integer
For i = TrendViewer30.enumCurves.cnCurve1 To
TrendViewer30.enumCurves.cnCurve3
       curve = m_Trend.GetCurve(i)
       data = curve.Data

       data.XValue = newXValue(i)
       data.YValue = newYValue(i)
       data.State = newState(i)
Next i
End If
```



Siehe auch

XGain, XOffset, YValue, YValueLong, AddXData, AddYData, State, Shared (TVXScale), Siehe auch Symbol,SymbolStep,TVLineStyle, TVStateStyle (TVCurve)

YGain

`Double` YGain()

Beschreibung

Definiert den Verstärkungsfaktor für den y-Wert einer Kurve.

Wertebereich

beliebiger Wert

Anmerkungen

Der Verstärkungsfaktor wird mit allen y-Werte der zugehörigen Kurve multipliziert. Die Standardeinstellung ist 1.

Eine genauere Erläuterung dieser Eigenschaft findet sich innerhalb der Beschreibung von **YOffset**.

Siehe auch

YValue, YValueLong, YOffset

YOffset

`Double` YOffset()

Beschreibung

Definiert den Aufsetzpunkt der Kurve für die y-Werte.

Wertebereich

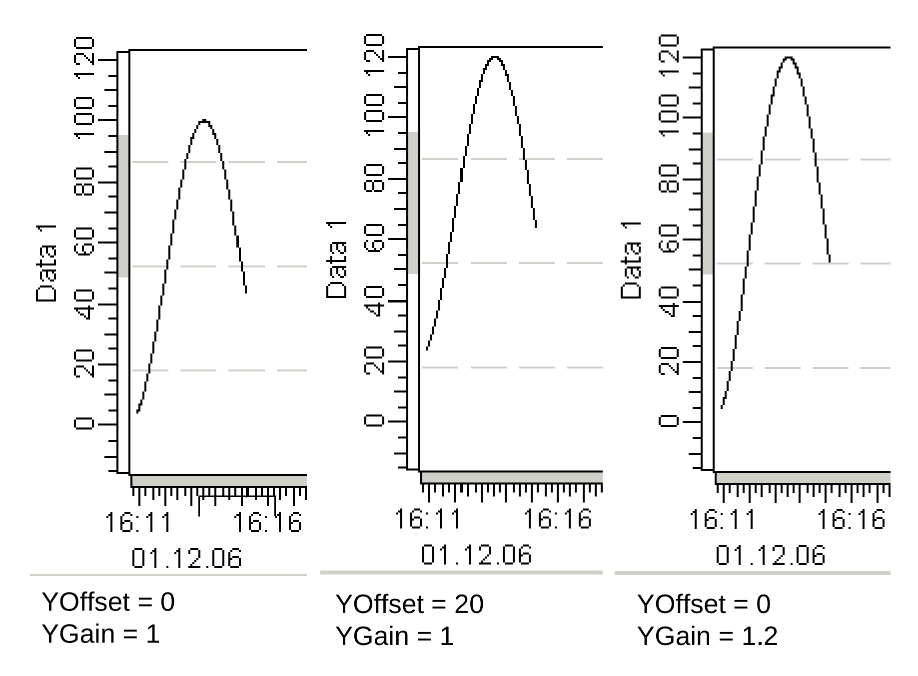
beliebiger Wert



Anmerkungen

Mit dem Festlegen des Aufsetzpunktes kann eine Kurve beliebig in y-Richtung nach oben bzw. nach unten verschoben werden. Die Standardeinstellung ist 0.

Folgende Abbildung soll die Einstellungen für Gain und Offset veranschaulichen:



Siehe auch

YValue, YValueLong, YGain

YValue, YValueLong

`Double` YValue()

`Long` YValueLong()

Beschreibung

Liefert den y-Wert des letzten Datenpunktes bzw. setzt den y-Wert eines neuen Datenpunktes.

Wertebereich

beliebiger Wert

Anmerkung

Eine genauere Erläuterung dieser Eigenschaft findet sich innerhalb der Beschreibung von **XValue, XValueLong, XValueDateTime**.

Siehe auch

YGain, YOffset, XValue, XValueLong, XValueDateTime



5.1.8 TVLineStyle

Eigenschaften des Linienstil-Objekts:

Color	Farbe der Linie
DashPattern	Benutzerdefinierter Linienstil
Style	Zeichenart der Linie (siehe **enumLineStyle**)
Width	Breite der Linie

Color

`System.Drawing.Color` Color()

Beschreibung

Definiert die Farbe der zu zeichnenden Linie.

Wertebereich

wird durch die Windowsfarbpalette vorgegeben

Siehe auch

Style, Width

DashPattern

`float[]` DashPattern()

Beschreibung

Legt einen benutzerdefinierten Linienstil fest.

Wertebereich

Wertearray mit Werten > 0

Anmerkung

Zum Umschalten in den Benutzerdefinierten Linienstil muß als **Style** Custom ausgewählt werden !

Programm-Beispiel

```
Dim Curve1 As TrendViewer30.CTVCurve
Dim linestyle1 As TrendViewer30.CTVLineStyle

Curve1 = m_Trend.GetCurve(0)
linestyle1 = Curve1.LineStyle

Dim styles(3) As Single
styles(0) = 4.0
styles(1) = 2.0
```



```
styles(2) = 5.0
styles(3) = 5.0

linestyle1.DashPattern = styles
```

Siehe auch

Style

Style

`System.Drawing.Drawing2D.DashStyle` Style()

Beschreibung

Definiert den Stil der Linie.

Wertebereich

siehe .Net `System.Drawing.Drawing2D.DashStyle`

Anmerkung

Das Zeichnen von Linien, die breiter als Strichstärke 1 (**Width**) sind, benötigen unter Umständen erheblich mehr Zeitaufwand.

Siehe auch

Color, Width, DashPattern, enumLineStyle

Width

`Short` Width()

Beschreibung

Definiert die Breite der zu zeichnenden Linie.

Wertebereich

`1`	Es wird eine dünne Linie gezeichnet (Standard-Einstellung)
`2...`	Es wird eine Linie der entsprechenden Dicke gezeichnet

Anmerkung

Das Zeichnen von Linien die breiter als Strichstärke 1 sind, benötigt unter Umständen erheblich mehr Zeitaufwand.

Siehe auch

Color, Style



5.1.9 TVFillStyle

Eigenschaften des Füllstil-Objekts:

Color	Definiert die Füllfarbe
HatchStyle	Füllstil
Solid	Legt fest, ob "ausgefüllt" gezeichnet werden soll

Color

`System.Drawing.Color` Color()

Beschreibung

Definiert die Füllfarbe.

Wertebereich

wird durch die Windowsfarbpalette vorgegeben

Siehe auch

HatchStyle

HatchStyle

`System.Drawing.Drawing2D.HatchStyle` Style()

Beschreibung

Definiert den Füllstil z.B. für Fluten, Hüllkurve.

Wertebereich

siehe .Net `System.Drawing.Drawing2D.HatchStyle`

Anmerkung

Je nach dem, welcher Füllstil ausgewählt wurde, benötigt unter Umständen erheblich mehr Zeitaufwand für das Zeichnen.

Siehe auch

Color

Solid

`boolean` Style()

Beschreibung

Legt fest, ob ausgefüllt gezeichnet, oder ein **HatchStyle** angewendet werden soll.

Wertebereich

siehe **enumFloodStyle**



Siehe auch

HatchStyle

5.1.10 TVStateStyle

Eigenschaften des Status-Objekts:

LineStyle	Referenz zu einem **TVLineStyle**-Objekt.
OnlyCurveColor	benutzt für den Status ausschließlich die Farbe der Kurve (`TRUE`)
State	Statuswert

LineStyle

`TVLineStyle` LineStyle()

Beschreibung

Referenz zu einem **TVLineStyle**-Objekt. Definiert den Linienstil der zu zeichnenden Statuslinie.

Wertebereich

siehe **TVLineStyle**

Anmerkungen

Zum Zeichnen von nicht sichtbaren Kurvenzügen muß der Alpha-Wert der Farbzuweisung auf den Wert 0 gesetzt werden.

```
LineStyle1.Color = Color.FromArgb(0,0,0,0)
```

Siehe auch

State, OnlyCurveColor

OnlyCurveColor

`Boolean` OnlyCurveColor()

Beschreibung

Verwendet beim Zeichnen der Statuslinie ausschließlich die Kurvenfarbe (**LineStyle**) (`TRUE`). Standardmäßig (`FALSE`) wird die Farbe der Statuslinie über **LineStyle** definiert.

Wertebereich

`TRUE`	Statuslinie wird in der Kurvenfarbe gezeichnet.
`FALSE`	Statuslinie wird in der Farbe gezeichnet, die in **LineStyle** definiert wird.



Anmerkungen

In der Regel wird die Farbe der Statuslinie über **LineStyle** definiert. Soll allerdings grundsätzlich die Kurvenfarbe **LineStyle (TVCurve)** zum Zeichnen verwendet werden, kann dies mit Einstellen dieser Eigenschaft vereinfacht werden.

Siehe auch

State, LineStyle, LineStyle (TVCurve)

State

`Integer` State()

Beschreibung

Diese Eigenschaft kann mit einem beliebigen ganzzahligen Wert definiert werden. Er dient als Referenz zum Statuswert, der in **State (TVData)** eingetragen wird.

Wertebereich

beliebiger ganzzahliger Wert

Programm-Beispiel

Eine genauere Erläuterung dieser Eigenschaft findet sich innerhalb der Beschreibung von **AddYData**.

Siehe auch

State (TVData), LineStyle,

5.1.11 TVRuler

Eigenschaften des Lineal-Objekts:

CustomStyle	aktiviert (`TRUE`) oder deaktiviert (`FALSE`) den benutzerdefinierten Linealstil
LineStyle	Referenz zu einem **TVLineStyle**-Objekt (Linenstil des benutzerdefinierten Lineals)
ViewType	Darstellungsart des Lineals (vertikal bzw. horizontal)
Visible	aktiviert (`TRUE`) oder deaktiviert (`FALSE`) die interaktive Steuerung des Lineals

---

CustomStyle

`Boolean` CustomStyle()

Beschreibung

Ein- bzw. Ausschalten des benutzerdefinierten Linealstils

Wertebereich

`TRUE`	Benutzerdefiniertes Lineal eingeschaltet
`FALSE`	Benutzerdefiniertes Lineal ausgeschaltet



Die Voreinstellung der Trend-Komponente verwendet das invertierende Standardlineal (**CustomStyle** = `FALSE`). Da dieses Lineal aber nicht in allen Geräte-Kontexten (Drucker beherrschen i. a. das Invertieren nicht) verwendet werden kann, erlaubt die Trend-Komponente auch die Verwendung eines benutzerdefinierten Lineals, dessen Aussehen über die Eigenschaften **LineStyle** bestimmt werden kann.

Siehe auch

LineStyle, ViewType, Visible

LineStyle

`TVLineStyle` LineStyle

Beschreibung

Legt den Linienstil des benutzerdefinierten (**CustomStyle**) Lineals fest.

Wertebereich

siehe **TVLineStyle**

Anmerkung

Diese Eigenschaft wird nur ausgewertet, wenn der benutzerdefinierte Linealstil aktiviert (**CustomStyle** = `TRUE`) ist.

Siehe auch

ViewType, CustomStyle, TVLineStyle

ViewType

`enumRulerStyle` ViewType()

Beschreibung

Legt die Darstellungsart (horizontal/vertikal) des Lineals fest.

Wertebereich

`0-`	Lineal ist versteckt
`1-`	vertikales Lineal
`2-`	horizontales Lineal

Anmerkung

Der Linealstil ist unabhängig von der Orientierung der Trend-Komponente (s. **Rotated**)

Siehe auch

LineStyle, Visible,



Visible

`Boolean` Visible()

Beschreibung

Legt fest, ob das Lineal angezeigt (sichtbar) werden sol.l

Wertebereich

`TRUE`	Lineal wird angezeigt
`FALSE`	Lineal wird nicht angezeigt

Siehe auch

ViewType

5.1.12 TVCrosshair

Eigenschaften des Lineal-Objekts:

Enabled	schaltet das Crosshair ein (`TRUE`) oder aus (`FALSE`)
Visible	Schaltet das Crosshair sichtbar (True) oder unsichtbar (False)

Enabled

`Boolean` Enabled()

Beschreibung

Ein- bzw. Ausschalten des Crosshairs

Wertebereich

`TRUE`	Crosshair eingeschaltet
`FALSE`	Crosshair ausgeschaltet

Siehe auch

Visible

Visible

`Boolean` Visible()

Beschreibung

Schaltet das Crosshair sichtbar (True) oder unsichtbar (False)

Wertebereich

`TRUE`	Crosshair sichtbar
`FALSE`	Crosshair nicht sichtbar

Siehe auch

Visible



5.1.13 TVSymbol

Eigenschaften des Symbol-Objekts:

Factor	Skalierungsfaktor für die Symbolgröße
Value	Legt die Identifikationsnummer des Symbols fest

Factor

`Double` Factor()

Beschreibung

Legt den Skalierungsfaktor für die Symbolgröße fest.

Wertebereich

beliebiger Gleitkommawert

Anmerkung

Der Faktorwert hat keinen Einfluss auf benutzerdefinierte Symbole !

Value

`Integer` Value()

Beschreibung

Legt die Identifikationsnummer für das Symbol (**enumCurveSymbols**) fest.

Wertebereich

`s.`**enumCurveSymbols**

Siehe auch

enumCurveSymbols



5.2 Methoden

Die folgende tabellarische Darstellung enthält eine Übersicht über alle Objekte der Trend-Komponente und deren Methoden:

Objekt-Gruppen:

5.2.1 TrendCtrl
5.2.2 TVXScale
5.2.3 TVYScale
5.2.4 TVPosition
5.2.5 TVCurve
5.2.6 TVData
5.2.7 TVRuler
5.2.8 TVCrosshair

5.2.1 TrendCtrl

Methoden der Trend-Komponente :

GetCurve	liefert das entsprechende **TVCurve**-Objekt zurück.
GetMarginFactor	liefert den Skalierungsfaktor für den Randbereich des Trends zurück.
GetTrendWindow	Ermittelt die Position des Trend-Fensters in der Trend-Komponente
GetXScale	liefert ein x-Skala-Objekt zurück.
GetYScale	liefert ein y-Skala-Objekt zurück.
PrintTrend	Druckt im Zielkontext einen Trend mit hoher Qualität
SetCursor	ändert die Maus-Cursor
SetMarginFactor	Legt die linke Randgröße der Trendfensters fest.
SetTrendWindow	setzt die Koordinaten des Trend-Fensters

GetCurve

`TVCurve` GetCurve(Short Index)

Beschreibung

Liefert das entsprechende **TVCurve**-Objekt zurück.

Wertebereich

siehe **TVCurve**

Siehe auch

TVCurve

GetMarginFactor

`Double` GetMarginFactor(`enumMargins` index)

Beschreibung

Liefert den Skalierungsfaktor für den Randbereich des Trends zurück.



Rückgabewert

Aktuelle Größe des Randbereichs

Siehe auch

TVXScale, TVYScale

GetTrendWindow

`Void` GetTrendWindow(ref `Integer` Left, ref `Integer` Top, ref `Integer` Width, ref `Integer` Height)

Beschreibung

Ermittelt die Position des Trend-Fensters in der Trend-Komponente.

Rückgabewert

keiner

Parameter

Left, Top, Width, Height	Position des Trend-Fensters in Pixel-Koordinaten der Trend-Komponente

Siehe auch

OwnerDrawUpdateCall, **SetPos**

GetXScale

`TVXScale` GetXScale(Integer Index)

Beschreibung

Liefert ein x-Skala-Objekt zurück.

Parameter

Index	Index der x-Skala

Rückgabewert

TVXScale

Anmerkung

Der "index" kann nur hier nur 0 oder 1 sein, da der *TrendViewer* über maximal 2 x-Skalen verfügt.

Siehe auch

TVYScale, TVXScale



GetYScale

`TVYScale` GetYScale(Integer Index)

Beschreibung

Liefert ein y-Skala-Objekt zurück.

Parameter

Index	Index der y-Skala

Rückgabewert

TVYScale

Anmerkungen

Der "index" sollte innerhalb der erzeugten y-Skalen-Objekte liegen (0..**YScales**). Die ersten 10 y-Skalen werden standardmäßig beim Anlegen der Trend-Komponente erzeugt. Es können zusätzlich bis zu 512 weitere y-Skalen erzeugt werden.

Siehe auch

TVXScale, TVYScale, YScales

PrintTrend

`Void` PrintTrend(`System.Drawing.Printing.PrintDocument` printDoc, Integer Left, Integer Top, Integer Width, Integer Height, `Bool` Rotate, `enumPrintLegendStyles` Legend)

Beschreibung

Druckt im Zielkontext einen Trend mit hoher Qualität.

Rückgabewert

keiner

Parameter

printDoc	Handle des Zielkontextes
Left, Top, Width, Height	Rechteck des Zielkontexts in logischen Pixelkoordinaten, in das der Trend gedruckt werden soll
Rotate	Druckt den Trend im rotierten Zustand (`TRUE`)
Legend	Stil für eine zusätzliche Legende zur Beschreibung der Kurven (s. **enumPrintLegendStyles**)

Programm-Beispiel

```
Dim destleft As Long, destwidth As Long, desttop As Long,_
         destheight As Long

destleft = 0: destwidth = 0: desttop = 0: destheight = 0

'start print
Printer.Print "" '"Trend-Print"
```



```
'we want pixels
Printer.ScaleMode = 3

'calculate destination frame
destleft = Printer.ScaleWidth / 10
destwidth = 6 * destleft
desttop = Printer.ScaleHeight / 10
destheight = 6 * desttop

'print destleft, desttop, destwidth, destheight, FALSE, _
         BOTTOM

TrendViewer1.PrintTrend Printer.hDC, destleft, desttop, _
         destwidth, destheight, 0, 3

Printer.EndDoc
```

Siehe auch

LockUpdate, enumPrintLegendStyles

`Void` PrintTrend(`System.Drawing.Graphics` aGraphics, Integer Left, Integer Top, Integer Width, Integer Height, `Bool` Rotate, `enumPrintLegendStyles` Legend)

Beschreibung

Druckt im Zielkontext einen Trend mit hoher Qualität.

Rückgabewert

keiner

Parameter

aGraphics	Handle des Zielkontextes
Left, Top, Width, Height	Rechteck des Zielkontexts in logischen Pixelkoordinaten, in das der Trend gedruckt werden soll
Rotate	Druckt den Trend im rotierten Zustand (`TRUE`)
Legend	Stil für eine zusätzliche Legende zur Beschreibung der Kurven (s. **enumPrintLegendStyles**)

Siehe auch

LockUpdate, enumPrintLegendStyles

SetCursor

`boolean` SetCursor(`enumMouseCursor` CursorIndex, `Cursor` NewCursor)

Beschreibung

Ändert den Maus-Cursor.



Wertebereich

`TRUE`	neuer Mauscursor übernommen
`FALSE`	neuer Mauscursor nicht übernommen

Parameter

CursorIndex	Index für die Cursorform, die geändert werden soll.
NewCursor	Neu zu setzender Cursors.

Programm-Beispiel

```
m_Trend.SetTrendCursor(enumMouseCursor.mcRulerHoriz,
Cursors.VSplit)
```

Siehe auch

enumMouseCursor

SetMarginFactor

`void` SetMarginFactor(`enumMargins` index, double NewValue)

Beschreibung

Legt den Skalierungsfaktor für den Randbereich des Trends fest.

Parameter

Index	Index des Randbereichs.
NewValue	Neue Größe des Randbereichs.

Rückgabewert

keiner

Siehe auch

GetMarginFactor

SetTrendWindow

`Void` SetTrendWindow(`Integer` Left, `Integer` Top, `Integer` Width, `Integer` Height)

Beschreibung

Setzt die Position des Trend-Fensters in der Trend-Komponente.

Rückgabewert

keiner

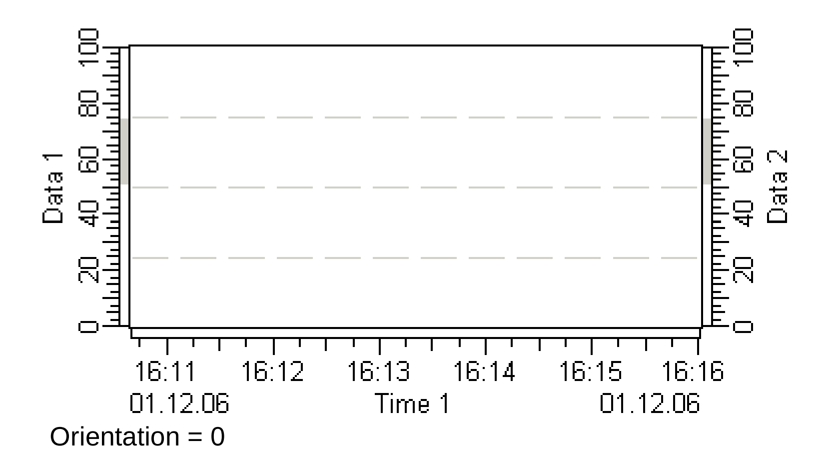
Parameter

Left, Top, Width, Height	Position des Trend-Fensters in Pixel-Koordinaten der Trend-Komponente

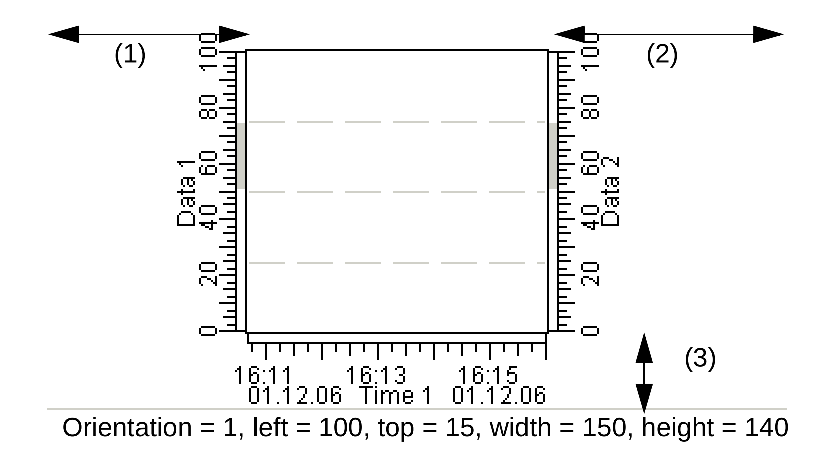


Anmerkungen

Diese Methode kann nur aufgerufen werden, wenn zuvor die Ausrichtung (**Orientation**) auf den Wert 1 (fixes Kurvenfenster) gesetzt wurde.



Orientation = 0



Orientation = 1, left = 100, top = 15, width = 150, height = 140

(1) Abstand zwischen Komponente links und Kurvenfenster links

(2) Abstand zwischen Komponente rechts und Kurvenfenster rechts

(3) Abstand zwischen Komponente unten und Kurvenfenster unten

Die Skalenbreite für alle Skalen ermittelt sich aus dem kleinsten Abstand der Komponente zum Kurvenfenster. Im Beispiel ist dies hier der Abstand (3). Alle Skalen werden somit mit der Breite (3) gezeichnet.

Siehe auch

Orientation, GetCurve



5.2.2 TVXScale

Methoden der x-Skala:

CalculatePoint	berechnet eine Fensterkoordinate für einen gegebenen x-Skalenwert
CalculateValue	berechnet einen x-Skalenwert zu einer bestimmten Fensterkoordinate
Capture	"Einfangen" aller Kurven auf der x-Koordinate
GetMarginFactor	Liefert den Skalenrand zurück
SetMarginFactor	Setzen eines Skalenrandes

CalculatePoint

`Boolean` CalculatePoint(`Double` Value, `Integer` pX,`Integer` pY)

Beschreibung

Berechnet die Fenster-Koordinate in Pixel, die dem übergebenen Skalenwert der jeweiligen x-Skala entspricht.

Rückgabewert

`TRUE`	wenn der Wert korrekt abgebildet werden konnte
`FALSE`	wenn der Wert außerhalb des maximalen Darstellungsbereichs (definiert durch **Range**) liegt

Parameter

Value	zu bearbeitender Skalenwert
pX, pY	Variablen, an die das Ergebnis übergeben wird

Programm-Beispiel

```
Dim xScale1 As TrendViewer30.CTVXScale
Dim pX As Integer, pY As Integer
Dim result As Boolean
xScale1 = m_Trend.GetXScale(0)
result = xScale1.CalculatePoint(DateTime.Now.ToOADate(),
pX, pY)
```

Siehe auch

CalculateValue, Range

CalculateValue

`Boolean` CalculateValue(`Integer` X, `Integer` Y, `Double` pValue)

Beschreibung

Berechnet zu einer beliebigen Fenster-Koordinate in Pixel den passenden Skalenwert der jeweiligen x-Skala.



Rückgabewert

`TRUE`	wenn die Fenster-Koordinate korrekt abgebildet werden konnte
`FALSE`	wenn der Ergebniswert außerhalb des maximalen Darstellungsbereichs (definiert durch **Range**) liegt

Parameter

X, Y	Fenster-Koordinate in Pixel
pValue	Variable, an die das Ergebnis übergeben wird

Programm-Beispiel

```
Dim xScale1 As TrendViewer30.CTVXScale
Dim pValue As Double
Dim result As Boolean
xScale1 = m_Trend.GetXScale(0)
result = xScale1.CalculateValue(100, 100, pValue)
```

Siehe auch

CalculatePoint, Range

Capture

`void` Capture()

Ermöglicht das "Einfangen" aller Kurven auf der x-Koordinate, die der jeweiligen x-Skala zugeordnet sind.

Anmerkung

Die Darstellung der Skala wird so angepaßt, das am linken Rand der kleinste x-Wert und am rechten Rand des Kurvenfensters der größte x-Wert aller Kurve(n) erscheint.

```
Dim xScale1 As TrendViewer30.CTVXScale
xScale1 = m_Trend.GetXScale(0)
xScale1.Capture()
```

GetMarginFactor

`Double` GetMarginFactor(`enumMargins` Index)

Beschreibung

Liefert die linke und rechte Randgröße einer x-Skala zurück.

Parameter

Index	linker oder rechter Skalenrand



Anmerkung

Die Skalenränder oben/unten werden bei den X-Skalen ignoriert !

Siehe auch

GetMarginFactor (TrendCtrl), Priority, TVYScale, SetMarginFactor

---

SetMarginFactor

`void` SetMarginFactor(`enumMargins` Index, `Double` Value)

Beschreibung

Legt die linke (Left) und rechte (Right) Randgröße der x-Skala fest.

Parameter

Index	linker oder rechter Skalenrand
Value	Größe des Skalenrandes

Anmerkung

Die Skalenränder oben/unten werden bei den X-Skalen ignoriert !

Siehe auch

GetMarginFactor (TrendCtrl), Priority, TVYScale,GetMarginFactor



5.2.3 TVYScale

Methoden des y-Skalenobjekts:

CalculatePoint	berechnet eine Fensterkoordinate für einen gegebenen x-Skalenwert
CalculateValue	berechnet einen x-Skalenwert zu einer bestimmten Fensterkoordinate
Capture	"Einfangen" aller Kurven auf der y-Koordinate
GetMarginFactor	Liefert den Skalenrand zurück
SetMarginFactor	Setzen eines Skalenrandes

CalculatePoint

`Boolean` CalculatePoint(`Double` Value, `Integer` pX, `Integer` pY)

Beschreibung

Berechnet die Fenster-Koordinate in Pixel, die dem übergebenen Skalenwert der jeweiligen x-Skala entspricht.

Rückgabewert

`TRUE`	wenn der Wert korrekt abgebildet werden konnte
`FALSE`	wenn der Wert außerhalb des maximalen Darstellungsbereichs (definiert durch **Range**) liegt.

Parameter

Value	zu bearbeitender Skalenwert
pX, pY	Variablen, an die das Ergebnis übergeben wird

Programm-Beispiel

```
Dim yScale1 As TrendViewer30.CTVYScale
Dim pX As Integer, pY As Integer
Dim result As Boolean
yScale1 = m_Trend.GetYScale(0)
result = yScale1.CalculatePoint(10.0, pX, pY)
```

Siehe auch

CalculateValue, Range

CalculateValue

`Boolean` CalculateValue(`Integer` X, `Integer` Y, `Double` pValue)

Beschreibung

Berechnet zu einer beliebigen Fenster-Koordinate in Pixel den passenden Skalenwert der jeweiligen x-Skala.



Rückgabewert

`TRUE`	wenn die Fenster-Koordinate korrekt abgebildet werden konnte
`FALSE`	wenn der Ergebniswert außerhalb des maximalen Darstellungsbereichs (definiert durch **Range**) liegt.

Parameter

X, Y	Fenster-Koordinate in Pixel
pValue	Variable, an die das Ergebnis übergeben wird

Programm-Beispiel

```
Dim yScale1 As TrendViewer30.CTVYScale
Dim pValue As Double
Dim result As Boolean
yScale1 = m_Trend.GetYScale(0)
result = yScale1.CalculateValue(100, 100, pValue)
```

Siehe auch

CalculatePoint, Range

Capture

`void` Capture()

Ermöglicht das "Einfangen" aller Kurven auf der y-Koordinate, die der jeweiligen y-Skala zugeordnet sind.

Anmerkung

Die Darstellung der Skala wird so angepaßt, das am unteren Rand der kleinste y-Wert und am oberen Rand des Kurvenfensters der größte y-Wert aller Kurve(n) erscheint.

Programm-Beispiel

```
Dim yScale1 As TrendViewer30.CTVYScale
yScale1 = m_Trend.GetYScale(0)
yScale1.Capture()
```

GetMarginFactor

`Double` GetMarginFactor(`enumMargins` Index)

Beschreibung

Liefert die obere oder untere Randgröße einer y-Skala zurück.

Parameter

Index	oberer oder unterer Skalenrand



Anmerkung

Die Skalenränder links/rechts werden bei den Y-Skalen ignoriert !

Siehe auch

GetMarginFactor (TrendCtrl), Priority, TVYScale, SetMarginFactor

---

SetMarginFactor

`void` SetMarginFactor(`enumMargins` Index, `Double` Value)

Beschreibung

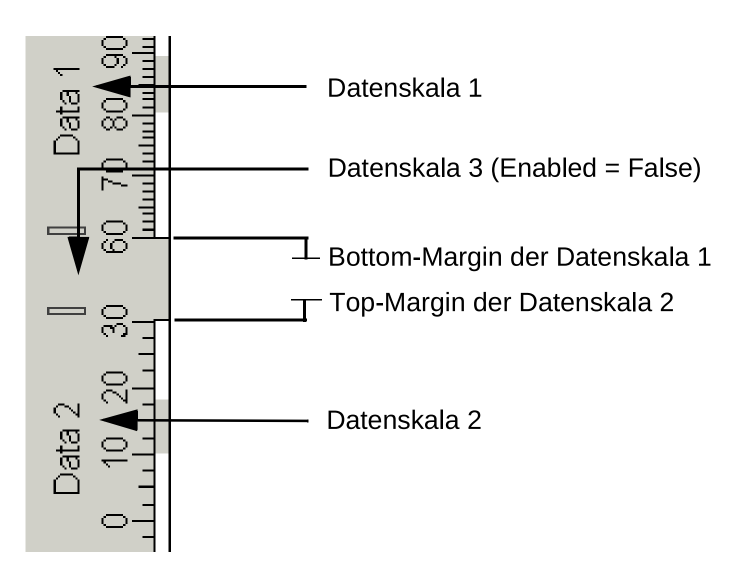
Legt die obere (Top) und untere (Bottom) Randgröße der y-Skala fest.

Anmerkungen

Diese Eigenschaft wird u.a. benötigt, um Überschneidungen der angrenzenden Y-Skalen zu vermeiden. Die Funktionalität betrifft ausschließlich übereinander geordneten Skalen.

Linker bzw. rechter Rand werden bei den Y-Skalen ignoriert.

Die folgende Abbildung veranschaulicht diese Eigenschaft:



Siehe auch

GetMarginFactor (TrendCtrl), Priority, TVYScale,GetMarginFactor



5.2.4 TVPosition

Methoden des Positionierungobjekts:

GetPosition	liefert die aktuelle Position einer Skala
SetPosition	setzt die aktuelle Position einer Skala

GetPosition

`Boolean` GetPosition(`Integer` pGroup, `Integer` pColumn, `Integer` pRow, `Double` pSize)

Beschreibung

Ermittelt die logische Position einer Skala.

Rückgabewert

`TRUE`	die gesuchte Skala wurde gefunden, die logische Position wurde übergeben
`FALSE`	die gesuchte Skala ist nicht aktiv (s. **Enabled TVXScale**, **Enabled TVYScale**).

Parameter

pGroup	Variable, welche die ermittelte Skalen-Gruppe empfangen soll (s. **enumScaleGroup**)
pColumn	Variable, welche die ermittelte Skalen-Spalte empfangen soll
pRow	Variable, welche die ermittelte Skalen-Reihe empfangen soll
pSize	Variable, welche die ermittelte Skalengröße in Prozent des Trend-Fensterrandes empfangen soll

Programm-Beispiel

siehe **SetPosition**

Siehe auch

Enabled (TVXScale), **Enabled (TVYScale)**, **SetPosition, enumScaleGroup**

SetPosition

`void` SetPosition(`Integer` Group, `Integer` `Column`, `Integer` Row, `Double` Size)

Beschreibung

Setzt die Skala auf eine neue logische Position.

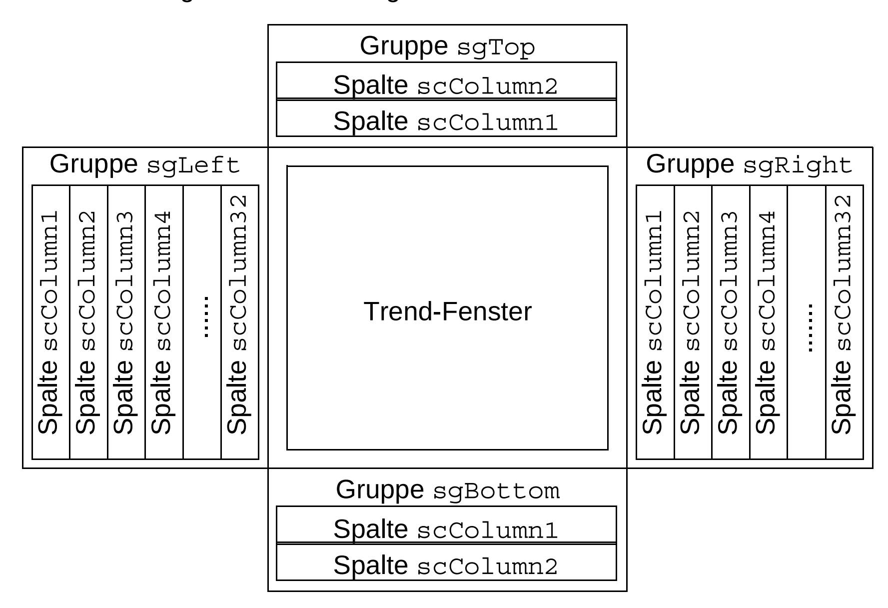


Parameter

Group	neue Skalen-Gruppe (s. **enumScaleGroup**)
Column	neue Skalen-Spalte
Row	neue Skalen-Reihe
Size	neue Skalen-Größe in Prozent des Trend-Fenster-randes

Anmerkungen

Die logische Position einer Skala wird durch ihre Gruppe, ihre Spalte, ihre Reihe und ihre relative Größe gegenüber dem Trend-Fensterrand bestimmt. Die Attribute der Skalenposition ist in der oben genannten Reihenfolge streng hierarchisch. Die Position der Gruppen und Spalten innerhalb der Komponente werden in der folgenden Zeichnung erläutert:



Jede Spalte besitzt mit Ausnahme der Spalten innerhalb der Gruppe `sgBottom u. sgTop` acht Reihen:

Spalte
Reihe `srRow1`
Reihe `srRow2`
Reihe `srRow3`
....
Reihe `srRow8`

Die Größe jeder Skala kann zwischen 0% und 100% liegen.

Die Komponente überprüft vor jedem Zeichenvorgang die Skalenpositionen und verändert gegebenfalls der Größe und Position, um die korrekte Position aller Skalen zu garantieren. Kann eine Skala nicht korrekt an ihrer Position plaziert werden, wird sie automatisch an ihre bisherige Position zurückgesetzt.



Die Komponente erlaubt keine Lücken innerhalb der einzelnen Skalenattribute und plaziert daher die Skalen u. Umständen in andere Positionen als die vorgegebenen, wie folgende Beispiele zeigen:

Beispiel 1: Nur die Spalten 1 und 2 innerhalb einer Gruppe sind besetzt, eine weitere Skala soll in Spalte 4 gesetzt werden, tatsächlich wird die Skala jedoch in Spalte 3 plaziert.

Beispiel 2: Sind die Spalten 1 bis 3 belegt und wird die Skala, die Spalte 2 als einzige belegt aus ihrer Position entfernt, werden die Skalen in Spalte 3 automatisch in Spalte 2 verschoben.

Beispiel 3: Nur die Reihen 1 und 2 sind innerhalb einer Spalte sind besetzt, eine weitere Skala soll in Reihe 4 gesetzt werden, tatsächlich wird die Skala jedoch in Reihe 3 plaziert.

Beispiel 4: Sind die Reihen 1 bis 3 belegt und wird die Skala in Reihe 2 aus ihrer Position entfernt, wird die Skala in Reihe 3 automatisch in Reihe 2 verschoben.

Wird eine Skala in eine Reihe plaziert, die bereits von einer anderen Skala belegt ist, so wandert diese Skala und alle Skalen in den folgenden Reihen um eine Reihe nach unten.

Wird eine Skala in eine bereits belegte Spalte plaziert und soll die neue Position in der ersten Reihe mit der Größe 100% liegen, so werden die Skalen in der entsprechenden Spalte und alle Skalen der folgenden Spalten um eine Spalte verschoben, so daß die neue Skala in die gewünschte Spalte eingefügt wird.

Beim Umpositionieren einer Skala muß berücksichtigt werden, daß eine bereits positionierte Skala vor der Neupositionierung zuerst entfernt wird, wie folgendes Beispiel zeigt:

Beispiel: Die Spalten 1 bis 3 sind belegt. Die Skala, welche die Spalte 1 vollständig belegt, soll nun in Spalte zwei positioniert werden. Beim Entfernen der Skala werden die Spalten 2 und 3 in die Spalten 1 und 2 verschoben. Anschließend wir dann die Skala in Spalte 2 eingefügt, wobei Spalte 2 nach Spalte 3 verschoben wird. Spalte 1 und Spalte 2 haben nach Beendigung des Vorgangs ihre Positionen getauscht.

Die folgenden Beispiele zeigen, wie man die Skalen korrekt plaziert.

Programm-Beispiel

```
Dim Position1 As TrendViewer30.CTVPosition
Dim Position2 As TrendViewer30.CTVPosition
Dim yScale1 As TrendViewer30.CTVYScale
Dim yScale2 As TrendViewer30.CTVYScale

yScale1 = m_Trend.GetYScale(0)
Position1 = yScale1.Position

yScale2 = m_Trend.GetYScale(1)
```



```
Position2 = yScale2.Position

Dim Group As Integer
Dim Row As Integer
Dim Col As Integer
Dim Size As Double

' vertausche Y-Skala 1 mit Y-Skala 2
Position1.GetPosition(Group, Col, Row, Size)
Group = TrendViewer30.enumScaleGroup.sgRight

Position1.SetPosition(Group, Row, Col, Size)
Group = TrendViewer30.enumScaleGroup.sgLeft

Position2.SetPosition(Group, Row, Col, Size)
```

Siehe auch

GetPosition, Group, Col, Row, Size

5.2.5 TVCurve

Methoden des Kurvenobjekts:

Capture	"Einfangen" der Kurve auf der x- und y-Koordinate
GetAuxValue	Liefert die Position einer Kurvenhilfslinien zurück.
GetAuxVisible	Liefert die Sichtbarkeit (TRUE) oder Unsichtbarkeit (FALSE) einer Kurvenhilfslienie zurück.
GetStateStyle	Liefert den Status des mit "index" übergebenen Styles.
SetAuxValue	Legt die Position einer Kurvenhilfslinien fest.
SetAuxVisible	Legt die Sichtbarkeit (TRUE) oder Unsichtbarkeit (FALSE) einer Kurvenhilfslinie fest.
SetStateStyle	Legt den Status des mit "index" übergebenen Styles fest.



Capture

`void` Capture()

Beschreibung

Ermöglicht das "Einfangen" einer Kurve. Alle Datenwerte (**DataMax, DataMin, DataMaxDateTime, DataMinDateTime**) einer Kurve werden im Trend-Fenster angezeigt.

Anmerkung

Die Darstellung der zugeordneten Skalen wird so angepaßt, das am oberen Rand der größte Y-Wert und am unteren der kleineste Y-Wert bzw. am linken Rand der kleinste X-Wert und am rechten Rand der größte X-Wert der Kurve erscheint.

Siehe auch

Data, DataMax, DataMin, DataMaxDateTime, DataMinDateTime

GetAuxValue

`Double` GetAuxValue(**enumCurveAuxLine** index)

Beschreibung

Liefert die Position einer Kurvenhilfslinien zurück.

Rückgabewert

Position der Hilfslinie

Parameter

index	Nummer der Kurvenhilfslinie, deren Wert zurückgeliefert werden soll

Anmerkungen

Die Hilfslinien der Kurven werden u.a. zur Markierung eines Wertebereichs für die Kurve verwendet. Ob eine Hilfslinie das Maximum bzw. das Minimum einer Bereichs markiert, wird durch kleine Symbole an den Hilfslinien gekennzeichnet. Die Symbole (das aktuelle Kurvensymbol) liegen immer im Wertebereich neben den Hilfslinien.

Siehe auch

AuxLineStyle, GetAuxVisible, enumCurveAuxLine

GetAuxVisible

`Boolean` GetAuxVisible(**enumCurveAuxLine** index)

Beschreibung

Liefert die Sichtbarkeit (TRUE) oder Ensichtbarkeit (FALSE) einer Kurvenhilfslienie zurück.



Rückgabewert

Hilflinie sichtbar (TRUE) oder unsichtbar (FALSE)

Parameter

index	Nummer der Kurvenhilfslinie

Siehe auch

AuxLineStyle, enumCurveAuxLine

GetStateStyle

`TVStateStyle` SetStateStyle(Integer index)

Beschreibung

Liefert den Status des mit "index" übergebenen Styles.

Wertebereich

siehe `TVStateStyle`

Anmerkungen

Jede Kurve kann eine fest angegebene Anzahl (**StateFromPoint**) an Statuswerten enthalten. Die verschiedenen Stauswerte können über **Data** und dem **TVData**-Objekt einem Kurvenpunkt zugeordnet werden. Beim Zeichnen der Kurve wird der Status und somit der Stil der Kurvenlinie solange gezeichnet, bis ein anderer Statuswert eintrifft.

Eine genauere Erläuterung dieser Eigenschaft findet sich innerhalb der Beschreibung von **AddYData**.

Siehe auch

StateFromPoint, Data, TVStateStyle, State (TVData), TVStateStyle, AddYData

SetAuxValue

`Void` SetAuxValue(**enumCurveAuxLine** index, Double value)

Beschreibung

Legt die Position einer Kurvenhilfslinien fest.

Rückgabewert

keiner

Parameter

index	Nummer der Kurvenhilfslinie, deren Wert gesetzt bzw. zurückgeliefert werden soll
value	Neue Position der Hilfsline



Siehe auch

AuxLineStyle, GetAuxVisible, enumCurveAuxLine

SetAuxVisible

`Void` SetAuxVisible(**enumCurveAuxLine** index, BOOL value)

Beschreibung

Definiert, ob die Hilfslinien(n) der jeweiligen Kurve angezeigt werden soll.

Rückgabewert

keiner

Parameter

index	Nummer der Kurvenhilfslinie, die angezeigt werden soll
value	sichtbar (TRUE) oder unsichtbar (FALSE)

Siehe auch

AuxLineStyle, enumCurveAuxLine

SetStateStyle

`TVLineStyle` SetStateStyle(Integer index)

Beschreibung

Legt den Status des mit "index" übergebenen Styles fest.

Rückgabewert

keiner

Parameter

index	Index des zu setzenden Styles

Siehe auch

StateFromPoint, Data, TVStateStyle, State (TVData), TVStateStyle, AddY-Data



5.2.6 TVData

Methoden des Kurvendatenobjekts:

AddXData	Fügt x-Werte zu einer Kurve hinzu
AddYData	Fügt y-Werte zu einer Kurve hinzu
DeleteData	löscht einen Datensatz
FindFirstXValue	sucht einen Wert in den x-Daten einer Kurve
FindFirstYValue	sucht einen Wert in den y-Daten einer Kurve
FindMinMaxXPos	sucht zwischen zwei Positionen nach dem kleinsten und größten x-Wert
FindMinMaxYPos	sucht zwischen zwei Positionen nach dem kleinsten und größten y-Wert
FindNextXValue	sucht ab einer bestimmten Stelle nach einem x-Wert
FindNextYValue	sucht ab einer bestimmten Stelle nach einem y-Wert
GetDataRange	liefert die aktuellen Wertebereiche einer Kurve
GetDataXCount	liefert die Anzahl aller x-Werte einer Kurve
GetDataYCount	liefert die Anzahl aller x-Werte einer Kurve
GetXData	liest einen Block von x-Werten aus
GetYData	liest einen Block von y-Werten aus
ReplaceXData	ersetzt einen Block von x-Werten
ReplaceYData	ersetzt einen Block von y-Werten
VisibleDataCount	Liefert die aktuelle sichtbare Anzahl an Kurvenpunkten des Datensatzes zurück, die im Trendfenster dargestellt werden

AddXData

`Integer` AddXData(`Double[]` xValues, `Integer` DataCount)

`Integer` AddXData(`Long[]` xValues, `Integer` DataCount)

`Integer` AddXData(`DateTime[]` xValues, `Integer` DataCount)

Beschreibung

Übergibt einen oder mehrere x-Werte für die Kurvenpunkte der jeweiligen Kurve an die Trend-Komponente.

Rückgabewert

keiner

Parameter

xValues	`Felder der Typen Double, Long` oder `DateTime`
DataCount	Anzahl der x-Werte, die übergeben werden sollen

Anmerkungen

Um die unterschiedlichen Datenmodelle, die durch die Eigenschaft **Shared** des **TVXScale**-Objekts gewählt werden zu unterstützen, müssen x- und y-Datenfelder getrennt übergeben werden. Um die Konsistenz der Datensätze zu gewährleisten, müssen die folgenden Regeln beachtet werden.



- Für **Shared** = `TRUE` darf nur einmal das entsprechende x-Datenfeld übergeben werden.
- Für **Shared** = `FALSE` muß für jede Kurve auch der x-Datensatz übergeben werden.

Während der Datenübergabe sollte kein Neuzeichnen der Komponente stattfinden (z.B. durch Abarbeitung der Windows-Messages zur Aktualisierung von Fenstern mit Fortschrittsbalken), da die Kurvendaten teilweise inkonsistent sind. Man sollte daher sicherheitshalber während dieser Zeit **LockUpdate** auf `TRUE` setzen.

Programm-Beispiel

siehe **AddYData**

Siehe auch

AddYData, LockUpdate, Shared, TVXScale

AddYData

`Integer` AddYData(`Double` YValues, `Integer` Status, Boolean bAddStatus, `Integer` DataCount)

`Integer` AddYData(`Long` YValues, `Integer` Status, Boolean bAddStatus, `Integer` DataCount)

Beschreibung

Übergibt einen oder mehrere y-Werte für die Kurvenpunkte der jeweiligen Kurve an die Trend-Komponente.

Der y-Wert einer Kurve wird aus einem gültigen Kurvenwert oder einem Fehlercode gebildet. Ist der Fehlercode ungleich 0, wird der übergebene y-Wert verworfen und durch den Fehlercode ersetzt.

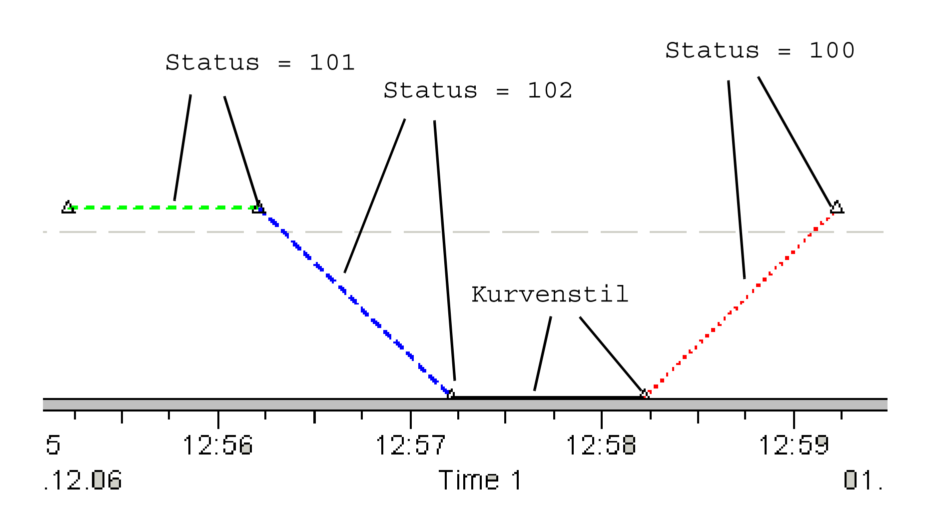
Rückgabewert

keiner

Parameter

YValues	Felder der Typen `Double` oder `Long`
Status	Feld des Typs `Integer` Statuswerte (s. **TVState-Style**)
bAddState	`TRUE`: es werden die übergebenen Statuswerte in den Datensatz integriert `FALSE`: die y-Werte werden ohne Statuswerte übergeben
DataCount	Anzahl der Datensätze, die übergeben werden sollen





Programm-Beispiel

```
Dim Data1 As TrendViewer30.CTVData
Dim Curve1 As TrendViewer30.CTVCurve
Dim xScale1 As TrendViewer30.CTVXScale
Dim Range1 As TrendViewer30.CTVRange
Dim Style1 As TrendViewer30.CTVStateStyle

m_Trend.LockUpdate = True

' 1. Curveobject
Curve1 = m_Trend.GetCurve(0)
Data1 = Curve1.data

' define 3 states
Curve1.StateStylesCount = 3

Style1 = Curve1.GetStateStyle(0)
Style1.State = 100

Dim LineStyle1 As TrendViewer30.CTVLineStyle

LineStyle1 = Style1.LineStyle
LineStyle1.Color = Color.Red
```



```
LineStyle1.Style = 2
LineStyle1.Width = 2

Style1 = Curve1.GetStateStyle(1)
Style1.State = 101

LineStyle1 = Style1.LineStyle
LineStyle1.Color = Color.Green
LineStyle1.Style = 1
LineStyle1.Width = 2

Style1 = Curve1.GetStateStyle(2)
Style1.State = 102

LineStyle1 = Style1.LineStyle
LineStyle1.Color = Color.Blue
LineStyle1.Style = 1
LineStyle1.Width = 2

' define range
xScale1 = Curve1.XScale
Range1 = xScale1.Range
Range1.ActualMaxDateTime = Now.AddHours(2.0)
Range1.ActualMinDateTime = Now.AddHours(-2)

' define Curvedata
Dim xArray(5) As DateTime
Dim yArray(5) As Double
Dim sArray(5) As Integer

Dim n As Date = Now

yArray(0) = 10
xArray(0) = n.AddMinutes(1.0)
sArray(0) = 100
```



```
yArray(1) = 10
xArray(1) = n.AddMinutes(2.0)
sArray(1) = 101

yArray(2) = 0
xArray(2) = n.AddMinutes(3.0)
sArray(2) = 102
' State not defined !
yArray(3) = 0
xArray(3) = n.AddMinutes(4.0)
sArray(3) = 4711

yArray(4) = 10
xArray(4) = n.AddMinutes(5.0)
sArray(4) = 100

Data1.AddXData(xArray, 5)
Data1.AddYData(yArray, sArray, True, 5)

' Kurvesymbol
Curve1.SymbolStep = 1

m_Trend.LockUpdate = False
```

Siehe auch

TVXScale, TVYScale, TVRange, TVCurve, TVStateStyle, TVData, Shared, LockUpdate

---

DeleteData

`Void` DeleteData()

Beschreibung

Löscht den Datensatz der Kurve.

Rückgabewert

keiner



Parameter

CurveNumber	Nummer der Kurve, deren Datensatz gelöscht werden soll

Anmerkungen

Ist **Shared** = `TRUE`, wird der x-Datensatz erst gelöscht, wenn alle y-Datensätze der aktiven Kurven gelöscht sind.

Das Löschen großer Datenmengen kann unter Umständen viel Zeit in Anspruch nehmen. In diesem Fall sollte die Applikation einen Fortschrittsbalken anzeigen. Dazu sollten die Datensätze unter Verwendung der Kurven-Eigenschaft **MaxDataSet** schrittweise verkleinert werden, bis zum Schluß durch **DeleteData** der Rest des Datensatzes gelöscht wird. **MaxDataSet** sollte anschließend auf den alten Wert zurückgesetzt werden.

Siehe auch

Shared, **MaxDataSet**

FindFirstXValue

`Integer` FindFirstXValue(`Double` xValue, `Double` Tolerance)

Beschreibung

Sucht den ersten x-Wert der Kurve, der gleich xValue mit der angegebenen Toleranz ist oder für eine negative Toleranz xValue überschreitet.

Rückgabewert

>0	Nummer des gefundenen Kurvenpunktes
<=0	nicht gefunden

Parameter

xValue		Zu suchender Wert
Tolerance	`>=0`:	Maximale Abweichung des zu suchenden Werts
	`<0`:	Sucht den ersten x-Wert, der xValue überschreitet

Anmerkung

Während der Arbeit mit dem Datensatz sollte dieser nicht durch hinzufügen von weiteren Datenpunkten geändert werden, da speziell für limitierte Datensatzlänge (s.**MaxDataSet**) Datensatzverschiebungen die Nummern der Datenpunkte zwischen zwei Suchvorgängen ändern kann.

Programm-Beispiel

siehe **FindFirstYValue**

Siehe auch

FindFirstYValue, MaxDataSet



FindFirstYValue

Integer FindFirstYValue(Double YValue, Double Tolerance)

Sucht den ersten y-Wert der Kurve, der gleich YValue mit der angegebenen Toleranz ist oder für eine negative Toleranz YValue überschreitet.

Rückgabewert

>0	Nummer des gefundenen Kurvenpunktes
<=0	nicht gefunden

Parameter

YValue		Zu suchender Wert
Tolerance	>=0:	Maximale Abweichung des zu suchenden Werts
	<0:	Sucht den ersten y-Wert, der YValue überschreitet

Anmerkung

Während der Arbeit mit dem Datensatz sollte dieser nicht durch Hinzufügen von weiteren Datenpunkten geändert werden, da speziell für limitierte Datensatzlänge (s.**MaxDataSet**) Datensatzverschiebungen die Nummern der Datenpunkte zwischen zwei Suchvorgängen ändern kann.

Programm-Beispiel

```
Dim xScale1 As TrendViewer30.CTVXScale
Dim Data1 As TrendViewer30.CTVData
Dim Curve1 As TrendViewer30.CTVCurve
Dim Range1 As TrendViewer30.CTVRange
Dim Ruler1 As TrendViewer30.CTVRuler

'find a data point and position the  near this point
'declaring variables

Dim xmin As Double, xmax As Double
Dim pXData() As Double = Nothing
Dim pYData() As Double = Nothing
Dim pStatus() As Integer = Nothing
Dim Position As Long

Curve1 = m_Trend.GetCurve(0)
Data1 = Curve1.data

'definition outside: Public Position As Long
```



```
'search for y-values of curve 2 passing 20.0

If (Position > 0) Then
          Position = Data1.FindNextYValue(Position, _
          20.0#, 1.0#)
End If
If (Position < 1) Then
    Position = Data1.FindFirstYValue(20.0#, 1.0#)
End If
If (Position < 1) Then
    MsgBox("No data found !")
Else
'we want the full data point
Data1.GetXData(Position, pXData, 1)
Data1.GetYData(Position, pYData, pStatus, False, 1)
'if the found value is not in the range of the actual
'display we will move the display, such that the
'data-point is near the center of the trend window
xScale1 = Curve1.XScale
Range1 = xScale1.Range
xmax = Range1.ActualMax()
xmin = Range1.ActualMin()

m_Trend.LockUpdate = True ' lock display
Range1.ActualMin() = pXData(0) - (xmax - xmin) / 2.0#
Range1.ActualMax() = pXData(0) + (xmax - xmin) / 2.0#
m_Trend.LockUpdate = False ' we want to see it

'we need the related point to set the  position
Dim pX As Object
Dim pY As Object
Dim res As Boolean
res = xScale1.CalculatePoint(pXData(0), pX, pY)

'set the ruler to the related point
```



```
Ruler1 = m_Trend.Ruler
Ruler1.SetPosition(pX, pY)
```

Siehe auch

MaxDataSet, FindFirstXValue, FindNextYValue, TVRuler, TVData, TVCurve, TVRange, TVXScale, TVYScale

---

FindMinMaxXPos

`Boolean` FindMinMaxXPos(`Long` Start, `Long` End, ref Long minPos, ref Long maxPos)

Sucht ab dem Kurvenpunkt mit der Nummer Start bis zum Kurvenpunkt mit der Nummer End nach dem kleinsten und dem größten x-Wert der Kurve.

true	Positionen wurden gefunden
false	Keine Position gefunden

Parameter

Start	Nummer des Kurvenpunktes, ab dem die Suche begonnen werden soll
End	Nummer des Kurvenpunktes, mit dem die Suche beendet werden soll
minPos	Zeiger auf die kleinste Position
maxPos	Zeiger auf die größte Position

Siehe auch

FindMinMaxYPos

FindMinMaxYPos

`Boolena` FindMinMaxYPos(`Long` Start, `Long` End, ref Long minPos, ref Long maxPos)

Sucht ab dem Kurvenpunkt mit der Nummer Start bis zum Kurvenpunkt mit der Nummer End nach dem kleinsten und dem größten y-Wert der Kurve.

Rückgabewert

true	Positionen wurden gefunden
false	Keine Position gefunden



Parameter

Start		Nummer des Kurvenpunktes, ab dem die Suche begonnen werden soll
End		Nummer des Kurvenpunktes, mit dem die Suche beendet werden soll
minPos		Zeiger auf die kleinste Position
maxPos		Zeiger auf die größte Position

Siehe auch

FindMinMaxXPos

FindNextXValue

`Integer` FindNextXValue(`Integer` Start, `Double` XValue, `Double` Tolerance)

Beschreibung

Sucht ab dem Kurvenpunkt mit der Nummer Start nach dem nächsten x-Wert der jeweiligen Kurve, der gleich XValue mit der angegebenen Toleranz ist oder für eine negative Toleranz XValue überschreitet.

Rückgabewert

>0	Nummer des gefundenen Kurvenpunktes
<=0	nicht gefunden

Parameter

Start		Nummer des Kurvenpunktes, ab dem die Suche begonnen werden soll
XValue		Zu suchender Wert
Tolerance	>=0:	Maximale Abweichung des zu suchenden Werts
	<0:	Sucht nächsten x-Wert, der XValue überschreitet

Programm-Beispiel

siehe **FindFirstYValue**

Siehe auch

FindFirstYValue

FindNextYValue

`Integer` FindNextYValue(`Integer` Start, `Double` YValue, `Double` Tolerance)

Beschreibung

Sucht ab dem Kurvenpunkt mit der Nummer Start nach dem nächsten y-Wert der jeweiligen Kurve, der gleich YValue mit der angegebenen Toleranz ist oder für eine negative Toleranz YValue überschreitet.



Rückgabewert

>0	Nummer des gefundenen Kurvenpunktes
<=0	nicht gefunden

Parameter

Start		Nummer des Kurvenpunktes, ab dem die Suche begonnen werden soll
YValue		Zu suchender Wert
Tolerance	>=0:	Maximale Abweichung des zu suchenden Werts
	<0:	Sucht nächsten y-Wert, der YValue überschreitet

Programm-Beispiel

siehe **FindFirstYValue**

Siehe auch

FindFirstYValue

GetDataRange

`Boolean` GetDataRange(`Double` pXMin, `Double` pXMax, `Double` pYMin, `Double` pYMax)

Beschreibung

Ermittelt die Wertebereiche der jeweiligen Kurve.

Rückgabewert

`TRUE`	Wertebereiche wurden übergeben
`FALSE`	die Kurve besitzt keinen Datensatz und somit keine gültigen Wertebereiche

Parameter

pXMin	Variable, welche das Minimum des x-Wertebereiches empfangen soll
pXMax	Variable, welche das Maximum des x-Wertebereiches empfangen soll
pYMin	Variable, welche das Minimum des y-Wertebereiches empfangen soll
pYMax	Variable, welche das Maximum des y-Wertebereiches empfangen soll

Programm-Beispiel

```
'fitting curve 1 to the trend window
Dim xRange1 As TrendViewer30.CTVRange
Dim yRange1 As TrendViewer30.CTVRange
Dim Curve1 As TrendViewer30.CTVCurve
Dim xScale1 As TrendViewer30.CTVXScale
Dim yScale1 As TrendViewer30.CTVYScale
```



```
Dim Data1 As TrendViewer30.CTVData
Dim pXMin As Double, pXMax As Double
Dim pYMin As Double, pYMax As Double

Curve1 = m_Trend.GetCurve(0)
Data1 = Curve1.data

xScale1 = Curve1.XScale
yScale1 = Curve1.YScale

xRange1 = xScale1.Range
yRange1 = yScale1.Range

Data1.GetDataRange(pXMin, pXMax, pYMin, pYMax)
m_Trend.LockUpdate = True

xRange1.ActualMax = pXMax
xRange1.ActualMin = pXMin

yRange1.ActualMax = pYMax
yRange1.ActualMin = pYMin


m_Trend.LockUpdate = False
```

Siehe auch

TVRange

GetDataXCount

`Integer` GetDataXCount()

Beschreibung

Ermittelt die Anzahl der x-Werte der jeweiligen Kurve.

Rückgabewert

Anzahl der x-Werte



Anmerkung

Diese Methode dient zur Überprüfung der Konsistenz der Datensätze.

Programm-Beispiel

siehe **ReplaceYData**

Siehe auch

ReplaceYData

GetDataYCount

`Integer` GetDataYCount()

Beschreibung

Ermittelt die Anzahl der y-Werte der jeweiligen Kurve.

Rückgabewert

Anzahl der y-Werte

Anmerkung

Diese Methode dient zur Überprüfung der Konsistenz der Datensätze.

Programm-Beispiel

siehe **ReplaceYData**

Siehe auch

ReplaceYData

GetXData

`Integer` GetXData(`Integer` Start, `ref Double[]` pXValues,
`Integer` DataCount)

`Integer` GetXData(`Integer` Start, `ref Long`[] pXValues, `Integer` DataCount)

`Integer` GetXData(`Integer` Start, `ref DateTime[]` pXValues,
`Integer` DataCount)

Beschreibung

Liest einen oder mehrere x-Datensätze der jeweiligen Kurve

Rückgabewert

Anzahl gelesener x-Datensätze

Parameter

Start	Nummer des Kurvenpunktes, ab dem die Daten gelesen werden sollen
pXValues	Referenz auf ein Datenfeld, das die x-Werte empfangen soll
DataCount	Anzahl zu lesender Kurvenpunkte



Programm-Beispiel

siehe **FindFirstYValue**

GetYData

`Integer` GetYData(`Integer` Start, `ref Double` pYValues, `ref Integer` Status, `Boolean` bGetStatus, `Integer` DataCount)

`Integer` GetYData(`Integer` Start, `ref Long` pYValues, `ref Integer` Status, `Boolean` bGetStatus, `Integer` DataCount)

Beschreibung

Liest einen oder mehrere Datensätze der jeweiligen Kurve.

Der y-Wert einer Kurve wird aus einem gültigen Kurvenwert oder einem Fehlercode gebildet. Ist der Fehlercode ungleich 0, wird der übergebene y-Wert verworfen und durch den Fehlercode ersetzt.

Rückgabewert

Anzahl gelesener Datensätze

Parameter

Start		Nummer des Kurvenpunktes, ab dem die Daten gelesen werden sollen
pYValues		Referenz auf ein Datenfeld, das die y-Werte empfangen soll
Status		Referenz auf ein Datenfeld, das die Fehlerwerte empfangen soll
bGetStatus	`TRUE`:	die Statuswerte werden in das übergebene Status-Feld eingetragen
	`FALSE`:	das übergebene Status-Feld wird nicht geändert (es kann ein beliebiger Zeiger übergeben werden)
DataCount		Anzahl zu lesender Kurvenpunkte

Programm-Beispiel

siehe **FindFirstYValue**

ReplaceXData

`Integer` ReplaceXData(`Integer` Start, `Double[]` xValues, `Integer` DataCount)

`Integer` ReplaceXData(`Integer` Start, `Long`[] xValues, `Integer` DataCount)

`Integer` ReplaceXData(`Integer` Start, `DateTime`[] xValues, `Integer` DataCount)

Beschreibung

Ersetzt die x-Werte der Kurvenpunkte der jeweiligen Kurve.

Rückgabewert

>0	Anzahl ersetzter Werte
<=0	ein Fehler ist aufgetreten



Parameter

CurveNumber	Nummer der Kurve, deren Daten ersetzt werden sollen
Start	Anfangs-Kurvenpunkt, ab dem die Daten ersetzt werden
xValues	Feld mit einem oder mehreren x-Werten der neuen Datenpunkte
DataCount	Anzahl der Kurvenpunkte, die ersetzt werden sollen

Anmerkungen

Die Methoden **ReplaceXData**, **ReplaceYData** können nur Werte in existierenden Datensätzen ersetzen. Es können keine neuen Datensätze hinzugefügt werden.

Ist **Shared** = `TRUE`, werden bei Austausch von x-Werten der Datenpunkte auch die x-Werte aller übrigen Kurven ausgetauscht.

Während der Datenübergabe sollte kein Neuzeichnen der Komponente stattfinden (z.B. durch Abarbeitung der Windows-Messages zur Aktualisierung von Fenstern mit Fortschrittsbalken), da die Kurvendaten teilweise inkonsistent sind. Man sollte daher sicherheitshalber während dieser Zeit **LockUpdate** auf `TRUE` setzen.

Programm-Beispiel

siehe **ReplaceYData**

Siehe auch

ReplaceYData, Shared, LockUpdate

ReplaceYData

`Integer` ReplaceYData(`Integer` Start, ref `Double[]` yValues, `VARIANT` Status, `Boolean` bAddStatus, `Long` DataCount)

`Integer` ReplaceYData(`Integer` Start, `ref Long[]` yValues, `VARIANT` Status, `Boolean` bAddStatus, `Integer` DataCount)

Beschreibung

Ersetzt die y-Werte der Kurvenpunkte einer Kurve.

Der y-Wert einer Kurve wird aus einem gültigen Kurvenwert oder einem Fehlercode gebildet. Ist der Fehlercode ungleich 0, wird der übergebene y-Wert verworfen und durch den Fehlercode ersetzt.

Rückgabewert

>0	Anzahl ersetzter Werte
<=0	ein Fehler ist aufgetreten



Parameter

Start		Anfangs-Kurvenpunkt, ab dem die Daten ersetzt werden
yValues		Feld mit einem oder mehreren zu ersetzenden y-Werte
Status		Feld mit einem oder mehreren zu ersetzenden Statuswerte
bAddStatus	`TRUE`:	wertet die übergebene Statis aus und trägt diese im Datensatz ein
	`FALSE`:	die ersetzten y-Werte enthalten keine Statuswerte
DataCount		Anzahl der Kurvepunkte, die ersetzt werden sollen

Programm-Beispiel

```
'replace ydata
'maximum 10000 points declarations
Dim Data1 As TrendViewer30.CTVData
Dim Curve1 As TrendViewer30.CTVCurve
Dim Data2 As TrendViewer30.CTVData
Dim Curve2 As TrendViewer30.CTVCurve
Dim pYData1() As Double, pYData2() As Double, pYData3() As
Double
Dim pStatus1() As Integer, pStatus2() As Integer,
pStatus3() As Integer
Dim MaxCount As Long, Count As Long

MaxCount = 10000

Curve1 = m_Trend.GetCurve(0)
Data1 = Curve1.data

Curve2 = m_Trend.GetCurve(1)
Data2 = Curve2.data

'calculate data to replace
Count = Data1.GetYDataCount()
If (MaxCount > Count) Then MaxCount = Count
Count = Data2.GetYDataCount()

If (MaxCount > Count) Then MaxCount = Count
```



```
m_Trend.LockUpdate = True

'get data to replace
Data1.GetYData(0, pYData3, pStatus3, True, MaxCount)
Data2.GetYData(0, pYData2, pStatus2, True, MaxCount)

'replace data
Data1.ReplaceYData(0, pYData2, pStatus2, True, MaxCount)
Data2.ReplaceYData(0, pYData3, pStatus3, True, MaxCount)

m_Trend.LockUpdate = False
```

Anmerkungen

Statuswerte werden nur berücksichtigt, wenn die Kurve zuvor bereits Statuswerte beinhaltet. Wurde die Kurve ohne Statuswerte initialisiert, werden übergebene Statuswerte ignoriert.

Siehe auch

TVYScale, TVCurve, TVData, GetDataYCount, LockUpdate

VisibleDataCount

`Boolean` VisibleDataCount(ref `Integer` minPos, ref `Integer` maxPos)

Beschreibung

Liefert die aktuelle sichtbare Anzahl an Kurvenpunkten des Datensatzes zurück, die im Trendfenster dargestellt werden.

Rückgabewert

True	Kurvenpunkte sind sichtbar
False	aktuell keine Kurvenpunkte sichtbar

Parameter

minPos	Position des ersten sichtbaren Kurvenpunktes
maxPos	Position des letzten sichtbaren Kurvenpunktes

Siehe auch

Average, Envelope, EnvelopeThreshold, AverageWithThreshold



5.2.7 TVRuler

Methoden des Linealobjekts:

GetCurveValue	Ermittelt den aktuellen Kurvenwert unter dem Lineal
GetPosition	Liefert die aktuelle Position des Lineal
GetScValue	Liefert den aktuellen Wert des Lineals in Bezug auf einer x-oder-y-Skala
SetPosition	Setzt das Lineal auf eine neue Position
SetScValue	Setzt das Lineal in Bezug auf den Wert einer x-oder-y-Skala an eine neue Position

---

GetCurveValue

`Boolean` GetCurveValue(**TVCurve** curve, ref `Double` pValue, `Short` Index)

Beschreibung

Ermittelt den aktuellen Kurvenwert unter dem Lineal.

Rückgabewert

`TRUE`	Kurvenwert konnte ermittelt werden
`FALSE`	Kurvenwert konnte nicht ermittelt werden

Parameter

curve	Kurve, von welcher der Wert gelesen werden soll
pValue	Referenz-Variable für den Kurvenwert.
Index	Reserviert für Erweiterung.

Siehe auch

TVCurve

GetPosition

`Boolean` GetPos(ref `Integer` pX, ref `Integer` pY)

Beschreibung

Ermittelt die aktuelle Position des Lineals.

Rückgabewert

`TRUE`	Linealposition konnte ermittelt werden
`FALSE`	Linealposition konnte nicht ermittelt werden

Parameter

pX	Variable, die die aktuelle x-Position des Lineals empfangen soll
pY	Variable, die die aktuelle y-Position des Lineals empfangen soll



Anmerkung

Wird als Position (-1,-1) zurückgeliefert, befindet sich das Lineal außerhalb des sichtbaren Bereiches.

Programm-Beispiel

```
Dim Ruler As TrendViewer30.CTVRuler
Dim pX As Integer, pY As Integer
Dim res As Boolean
Ruler = m_Trend.Ruler
res = Ruler.GetPosition(pX, pY)
```

Siehe auch

SetPosition

```
Dim Ruler As TrendViewer30.CTVRuler
Dim pX As Integer, pY As Integer
Dim res As Boolean
Ruler = m_Trend.Ruler
res = Ruler.GetPosition(pX, pY)
res = Ruler.SetPosition(0, 300)
```

Siehe auch

GetCurveValue

GetScValue

`Double` GetScValue(**TVXScale** Scale, `Boolean IsInRange, Short` Index)

`Double` GetScValue(**TVYScale** Scale, `Boolean IsInRange, Short` Index)

Beschreibung

Ermittelt die aktuelle Position des Lineals auf einer Skala.

Rückgabewert

`Double`	Wert auf der übergebenen Skala

Parameter

Scale	Referenz zu einer x-oder y-Skala (**TVXScale**, **TVYScale**)
IsInRange	1 = im Bereich, 0 = nicht im Bereich
Index	Reserviert für Erweiterung ! (=0)



Programm-Beispiel

```
Dim Ruler As TrendViewer30.CTVRuler
Dim xScale1 As TrendViewer30.CTVXScale
Dim Value As Double
Dim bInRange As Booelan
xScale1 = m_Trend.GetXScale(0)
Ruler = m_Trend.Ruler
Value = Ruler.GetXScValue(xScale1, bInRange, 0)
```

Siehe auch

SetScValue, SetPosition, TVXScale, **TVYScale**

SetPosition

`Boolean` SetPos(`Integer` pX, `Integer` pY)

Beschreibung

Setzt das Lineal an die angegebene Position.

Rückgabewert

`TRUE`	Linealposition konnte gesetzt werden
`FALSE`	Linealposition konnte nicht gesetzt werden

Parameter

pX	Variable, die die aktuelle x-Position des Lineals enthalten soll
pY	Variable, die die aktuelle y-Position des Lineals enthalten soll

Siehe auch

GetCurveValue

SetScValue

`Void` SetScValue(**TVXScale** Scale, `Double` Value, `Short` Index)

`Void` SetScValue(**TVYScale** Scale, `Double` Value, `Short` Index)

Beschreibung

Setzt das Lineal an die Position des Wertes "Value" auf der übergebenen Skala.



Parameter

Scale	Referenz zu einer x-oder y-Skala (**TVXScale**, **TVYScale**)
Value	Wert auf der Skala, auf welchen das Lineal positioniert werden soll
Index	Reserviert für Erweiterung ! (=0)

Programm-Beispiel

```
Dim Ruler As TrendViewer30.CTVRuler
Dim xScale1 As TrendViewer30.CTVXScale
xScale1 = m_Trend.GetXScale(0)
Ruler = m_Trend.Ruler
Ruler.SetXScValue(xScale1, Now.ToOADate(), 0)
```

Siehe auch

GetScValue, SetPosition, TVXScale, **TVYScale**

5.2.8 TVCrosshair

Methoden des Crosshairobjekts:

GetCurvePos	Ermittelt den aktuellen Kurvenwert unter dem Lineal
GetPosition	Liefert den aktuellen Wert des Lineals in Bezug auf einer x-oder-y-Skala
GetScValue	Liefert den aktuellen Wert des Lineals in Bezug auf einer x-oder-y-Skala

GetCurvePos

`Integer` GetCurvePos(**TVCurve** curve)

Beschreibung

Ermittelt den aktuellen Kurvenwert unter dem Crosshair.

Rückgabewert

`-1`	Kurvenwert konnte nicht ermittelt werden
`>=0`	aktueller Kurvenwert

Parameter

curve	Kurve, von welcher der Wert gelesen werden soll



Siehe auch

TVCurve

GetPosition

`Boolean` GetPos(ref `Integer` pX, ref `Integer` pY)

Beschreibung

Ermittelt die aktuelle Position des Crosshairs.

Rückgabewert

`TRUE`	Position konnte ermittelt werden
`FALSE`	Position konnte nicht ermittelt werden

Parameter

pX	Variable, die die aktuelle x-Position des Crosshair empfangen soll
pY	Variable, die die aktuelle y-Position des Crosshair empfangen soll

Anmerkung

Wird als Position (-1,-1) zurückgeliefert, befindet sich das Crosshair außerhalb des sichtbaren Bereiches.

Siehe auch

GetCurvePos

GetScValue

`Double` GetScValue(**TVXScale** Scale, `Short` Index)

`Double` GetScValue(**TVYScale** Scale, `Short` Index)

Beschreibung

Ermittelt die aktuelle Position des Crosshairs auf einer Skala.

Rückgabewert

`Double`	Wert auf der übergebenen Skala

Parameter

Scale	Referenz zu einer x-oder y-Skala (**TVXScale**, **TVYScale**)
Index	Reserviert für Erweiterung ! (=0)



Siehe auch

TVXScale, **TVYScale**



5.3 Ereignisse

Die folgende tabellarische Darstellung enthält eine Übersicht über alle Ereignisse der Trend-Komponente:

Event Groups

Events

5.3.1 Änderungs-Events

CrosshairPosChanged	wird verschickt, wenn die Position des Crosshairs verändert wurde
CurveAuxChanged	wird verschickt, wenn die Position einer Kurvenhilfslinie verändert wurde.
CurveChanged	wird verschickt, wenn eine Kurvetaste betätigt wurde
CurveMoved	wird verschickt, wenn die Position eines Kurvenpunktes bzw. die Position der Kurve verändert wurde
CurveYScaleChanged	wird verschickt, wenn die Y-Skalenzuordnung einer Kurve verändert wurde
RulerPosChanged	wird verschickt, wenn die Position eines Kurvenlineals verändert wurde
XScActualRangeChanged	wird verschickt, nach dem der Anzeigebereich der x-Skala verändert wurde
XScActualRangeChanging	wird verschickt, wenn der Anzeigebereich der x-Skala verändert wird
XScaleChanged	wird verschickt, wenn ein Attribut der x-Skala verändert wurde
XScalePosChanged	wird verschickt, wenn die Position der x-Skala verändert wurde
YScActualRangeChanging	wird verschickt, wenn der Anzeigebereich der y-Skala verändert wird
YScalePosChanged	wird verschickt, wenn die Position der y-Skala verändert wurde

CrosshairPosChanged

CrosshairPosChanged(**TVCrosshair** Crosshair)

Beschreibung

Tritt auf, nach dem der Benutzer die Position des Crosshairs verändert hat.

Parameter

Crosshair	zugehöriges Crosshair-Objekt

Siehe auch

TVCrosshair



CurveAuxChanged

CurveAuxChanged(**TVCurve** Curve)

Beschreibung

Tritt auf, nach dem der Benutzer die Position einer Kurvenhilfslinie verändert hat.

Parameter

Curve	zugehöriges Kurvenobjekt der Hilfslinie

Siehe auch

TVCurve

CurveChanged

CurveChanged(**TVCurve** Curve)

Beschreibung

Tritt auf, wenn der Benutzer eine Kurventaste betätigt.

Parameter

Curve	zugehöriges Kurvenobjekt

Siehe auch

TVCurve, CurveMoved

CurveMoved

CurveMoved(**TVCurve** Curve, `Double` oldX, `Double` oldY, `Double` actualX, `Double` actualY, Integer movePoint)

Beschreibung

Tritt auf, wenn der Benutzer die Position eines Kurvenpunktes bzw. die Position der Kurve verändert hat.

Parameter

Curve	zugehöriges Kurvenobjekt
oldX	alter x-Kurvenwert
oldY	alter y-Kurvenwert
actualX	aktueller x-Kurvenwert
actualY	aktueller y-Kurvenwert
movePoint	Wenn ein Kurvenpunkt verschoben wurde, enthält dieser Wert den Index des Kurvenpunktes, ansonsten -1

Siehe auch

TVCurve, CurveChanged



CurveYScaleChanged

CurveYScaleChanged(**TVCurve** Curve)

Beschreibung

Tritt auf, wenn der Benutzer die Skalenzuordnung einer Curve verändert hat.

Parameter

Curve	zugehöriges Kurvenobjekt

Siehe auch

TVCurve, TVYScale

RulerPosChanged

RulerPosChanged(**TVRuler** Ruler, **TVCurve** Curve)

Beschreibung

Tritt auf, wenn der Benutzer die Position eines zu einer Kurve zugeordneten Lineals verändert hat.

Parameter

Ruler	Referenz zu einem Linealobjekt
Curve	zugehöriges Kurvenobjekt

Siehe auch

TVCurve, **TVRuler**

XScActualRangeChanged

XScActualRangeChanged(**TVXScale** xScale, `Double` Min, `Double` Max)

Beschreibung

Dieser Event wird nach einer Änderung eines aktuell sichtbaren x-Skalenbereiches gefeuert.

Parameter

xScale	Referenz der x-Skala
Min	Wert der Eigenschaft (**ActualMax, ActualMin, ActualMaxDateTime, ActualMinDateTime**)
Max	Wert der Eigenschaft (**ActualMax, ActualMin, ActualMaxDateTime, ActualMinDateTime**)

Siehe auch

TVXScale, **ActualMax, ActualMin, ActualMaxDateTime, ActualMinDateTime**



XScActualRangeChanging

XScActualRangeChange(**TVXScale** xScale, `Double` Min, `Double` Max)

Beschreibung

Dieser Event wird bei jeder Änderung eines aktuell sichtbaren x-Skalenbereiches gefeuert.

Parameter

xScale	Referenz der x-Skala
Min	Wert der Eigenschaft (**ActualMax, ActualMin, ActualMaxDateTime, ActualMinDateTime**)
Max	Wert der Eigenschaft (**ActualMax, ActualMin, ActualMaxDateTime, ActualMinDateTime**)

Siehe auch

TVXScale, **ActualMax, ActualMin, ActualMaxDateTime, ActualMinDateTime**

XScaleChanged

XScaleChanged(**TVXScale** xScale)

Beschreibung

Tritt auf, wenn sich der Stil oder ein Wertebereich der x-Skala ändert.

Parameter

xScale	Referenz der x-Skala

Siehe auch

TVXScale, XScalePosChanged

XScalePosChanged

XScaleChanged(**TVXScale** xScale)

Beschreibung

Tritt auf, nach dem sich die Position der x-Skala geändert hat.

Parameter

xScale	Referenz der x-Skala

Siehe auch

TVXScale, XScaleChanged



YScActualRangeChanged

YScActualRangeChanged(**TVYScale** yScale, `Double` Min, `Double` Max)

Beschreibung

Dieser Event wird nach einer Änderung eines aktuell sichtbaren y-Skalenbereiches gefeuert.

Parameter

yScale	Referenz der y-Skala
Min	Wert der Eigenschaft (**ActualMax, ActualMin, ActualMaxDateTime, ActualMinDateTime**)
Max	Wert der Eigenschaft (**ActualMax, ActualMin, ActualMaxDateTime, ActualMinDateTime**)

Siehe auch

TVYScale, ActualMax, ActualMin, ActualMaxDateTime, ActualMinDateTime, XScActualRangeChanged

YScActualRangeChanging

YScActualRangeChange(**TVYScale** yScale, `Double` Min, `Double` Max)

Beschreibung

Dieser Event wird bei jeder Änderung eines aktuell sichtbaren y-Skalenbereiches gefeuert.

Parameter

yScale	Referenz der y-Skala
Min	Wert der Eigenschaft (**ActualMax, ActualMin, ActualMaxDateTime, ActualMinDateTime**)
Max	Wert der Eigenschaft (**ActualMax, ActualMin, ActualMaxDateTime, ActualMinDateTime**)

Siehe auch

TVYScale, ActualMax, ActualMin, ActualMaxDateTime, ActualMinDateTime, XScActualRangeChanged

YScaleChanged

YScaleChanged(**TVYScale** yScale)

Beschreibung

Tritt auf, wenn sich der Stil oder ein Wertebereich der jeweiligen y-Skala ändert.

Parameter

yScale	Referenz der y-Skala

Siehe auch

TVYScale, YScalePosChanged



YScalePosChanged

YScaleChanged(**TVYScale** yScale)

Des

Tritt auf, nach dem sich die Position der x-Skala geändert hat.

Parameter

yScale	Referenz der y-Skala

Siehe auch

TVYScale, YScActualRangeChanging

5.3.2 Zeichen-Events

PostDrawCurves	wird verschickt, bevor die Kurven gezeichnet werden
PreDrawCurves	wird verschickt, nachdem die Kurven gezeichnet wurden
XScaleTickText	wird verschickt, bevor ein Ticktext einer X-Skala gezeichnet wird
YScaleTickText	wird verschickt, bevor ein Ticktext einer Y-Skala gezeichnet wird

PostDrawCurve

PostDrawCurve(`Graphics` hDC, **TVCurve** `curve, Integer` Left, `Integer` Top, `Integer` Width, `Integer` Height);

Beschreibung

Wird innerhalb des Zeichenvorgangs verschickt, nachdem die referenzierte Kurve gezeichnet wurde.

Parameter

Graphics	Grafik-Geräte-Kontexts, in welchem der Trend gezeichnet wird
curve	referenziertes **TVCurve** - Objekt
Left, Top, Width, Height	Koordinaten des Bereichs des Zielfensters (in Pixel), der neu gezeichnet werden muß.

Anmerkung

Dieser Event kann dazu benutzt werden, in den Vordergrund des Trend-Fensters zu zeichnen. Auch während des Ausdrucks wird dieser Event verschickt.

Siehe auch

PreDrawCurve, PreDrawCurve, PreDrawCurves, TVCurve



PostDrawCurves

PostDrawCurves(`Graphics` hDC, `Integer` Left, `Integer` Top, `Integer` Width, `Integer` Height);

Beschreibung

Wird innerhalb des Zeichenvorgangs verschickt, nachdem das Trend-Fenster gezeichnet wurde.

Parameter

Graphics	Grafik-Geräte-Kontexts, in welchem der Trend gezeichnet wird
Left, Top, Width, Height	Koordinaten des Bereichs des Zielfensters (in Pixel), der neu gezeichnet werden muß.

Anmerkung

Dieser Event kann dazu benutzt werden, in den Vordergrund des Trend-Fensters zu zeichnen. Auch während des Ausdrucks wird dieser Event verschickt.

Siehe auch

PreDrawCurve, PostDrawCurve, PreDrawCurves, TVCurve

PreDrawCurve

PreDrawCurve(`Grahpic` hDC, **TVCurve** `curve`, `Integer` Left, `Integer` Top, `Integer` Width, `Integer` Height);

Beschreibung

Wird innerhalb des Zeichenvorgangs verschickt, nachdem der Trend-Hintergrund gezeichnet wurde und bevor die referenzierte Kurve gezeichnet wird.

Parameter

Graphic	Grafik-Geräte-Kontextes, in welchem der Trend gezeichnet wird
curve	referenziertes **TVCurve** - Objekt
Left, Top, Width, Height	Koordinaten des Bereichs des Zielfensters (in Pixel), der neu gezeichnet werden muß.

Anmerkung

Dieser Event kann dazu benutzt werden, in den Hintergrund des Trend-Fensters zu zeichnen. Auch während des Ausdrucks wird dieser Event verschickt.

Siehe auch

PostDrawCurves, PostDrawCurve, PreDrawCurves, TVCurve



PreDrawCurves

PreDrawCurves(`Grahpic` hDC, `Integer` Left, `Integer` Top, `Integer` Width, `Integer` Height);

Beschreibung

Wird innerhalb des Zeichenvorgangs verschickt, nachdem der Trend-Hintergrund gezeichnet wurde und bevor die Kurven gezeichnet werden.

Parameter

Graphic	Grafik-Geräte-Kontextes, in welchem der Trend gezeichnet wird
Left, Top, Width, Height	Koordinaten des Bereichs des Zielfensters (in Pixel), der neu gezeichnet werden muß.

Anmerkung

Dieser Event kann dazu benutzt werden, in den Hintergrund des Trend-Fensters zu zeichnen. Auch während des Ausdrucks wird dieser Event verschickt.

Siehe auch

PostDrawCurves, PostDrawCurve, PreDrawCurve, TVCurve

XScaleTickText

XScaleChanged(**TVXScale** xScale, `Integer` pos, `ref String` strText)

Beschreibung

Wird innerhalb des Zeichenvorgangs verschickt, bevor die Beschriftung eines Ticktextes erfolgt.

Parameter

xScale	Referenz der x-Skala
pos	Tickposition innerhalb der Skala
strText	Ticktext

Siehe auch

TVXScale

YScaleTickText

XScaleChanged(**TVYScale** yScale, `Integer` pos, `ref String` strText)

Beschreibung

Wird innerhalb des Zeichenvorgangs verschickt, bevor die Beschriftung eines Ticktextes erfolgt.



Parameter

yScale	Referenz der y-Skala
pos	Tickposition innerhalb der Skala
strText	Ticktext

Siehe auch

TVXScale



5.4 Deklarationen

Alle von *TrendViewer* verwendete Konstanten werden durch "enum"-Typen definiert, die im folgenden Abschnitt genauer erläutert werden.

"enum"-Typen:

enumAuxLineTyp	Enum-Typ für die Ausrichtung der Hilfsline
enumCurveAuxLine	Enum-Typ für die Hilfslinie
enumCurveButtons	Enum-Typ für die Position der Kurvenknöpfe
enumCurves	Enum-Typ für die Kurvennummern
enumCurveSymbols	Enum-Typ für die Kurvensymbole
enumCurveTypes	Enum-Typ für die Kurventypen
enumFloodStyle	Enum-Typ für die Füllstile
enumLineStyle	Enum-Typ für die Linienstile
enumMargins	Enum-Typ für die Randbereiche
enumPrintLegendStyles	Enum-Typ für die Position der Legende beim Ausdruck
enumRulerStyle	Enum-Typ für die Hilfslineale
enumStripChartsStyle	Enum-Typ für die Stripchart-Stile
enumScaleGroup	Enum-Typ für die Gruppennummer der Skalenpositionierung
enumScaleTypes	Enum-Typ für die Skalentypen
enumTextAlign	Enum-Typ für die Textausrichtung in der Skalenlegende
enumMouseCursor	Enum-Typ für die verschiedenen Maus-Cursor eines Trends
enumTrendZoom	Enum-Typ für die Zoomvarianten
enumXScalePosRows	Enum-Typ für die Zeilennummern der x-Skalenpositionen
enumXScalePosColumn	Enum-Typ für die Spaltennummern der x-Skalenpositionen
enumXScales	Enum-Typ für die Bezeichner der x-Skalen
enumYScalePosRows	Enum-Typ für die Zeilennummern der y-Skalenpositionen
enumYScalePosColumn	Enum-Typ für die Spaltennummern der y-Skalenpositionen
enumYScales	Enum-Typ für die Bezeichner der y-Skalen



enumAuxLineTyp

Defniert die Ausrichtung der Hilfslinien.

```
enum enumAuxLineTyp
{
 tlInvalid = -1,   // keine Zuordnung
 tlClassic = 0,    // klassische Ausrichtung
 tlGrid = 1,       // Ausrichtung an den Grids einer Skala
}
```

enumCurveAuxLine

Defniert die Hilfslinie.

```
enum enumCurveAuxLine
{
 crvAuxLineFirst = 0, // erste Kurvenhilfslinie
 crvAuxLineSecond = 1,// zweite Kurvenhilfslinie
}
```

enumCurveButtons

Defniert alle möglichen Positionen der Kurventasten.

```
enum enumCurveButtons
{
 cbNone = 0,       // keine Kurventasten
 cbTop = 1,        // Kurventasten sind oben
 cbScale = 2,      // Kurventasten sind an den Y-Skalen
}
```

enumCurves

Definiert die Identifikationsnummer für die Kurven.

```
enum enumCurves
{
 cnCurve1 = 0,        // 1. Kurve
 cnCurve2 = 1,        // 2. Kurve
 cnCurve3 = 2,        // 3. Kurve
 cnCurve4 = 3,        // 4. Kurve
 cnCurve5 = 4,        // 5. Kurve
 cnCurve6 = 5,        // 6. Kurve
 cnCurve7 = 6,        // 7. Kurve
 cnCurve8 = 7,        // 8. Kurve
 cnCurve9 = 8,        // 9. Kurve
 cnCurve10 = 9,       // 10. Kurve
 cnCurveMax = 10,     // 511. Kurve (letzte mögl. Kurve)
}
```



enumCurveSymbols

Definiert die Identifikationsnummer für die Kurvensymbole.

```
enum enumSymbols
{
 tcTriangle = 0,   // Dreieck
 tcQuad = 1,       // Quadrat
 tcRhomb = 2,      // Raute
 tcOct = 3,        // Oktaeder
 tcCross = 4,      // Kreuz
 tcX = 5,          // X
 tcDCross = 6,     // Doppelkreuz
 tcQuadCross = 7,  // Viereck mit Kreuz
 tcQuadX = 8,      // Viereck mit X
 tcRhombCross = 9, // Raute mit Kreuz
 tcOctCross = 10,  // Oktaeder mit Kreuz
 tcOctX = 11,      // Oktaeder mit X
}
```

enumCurveTypes

Definiert die verschiedenen Kurventypen.

```
enum enumCurveTypes
{
 isLinear = 0,        // linear
 isStepsRight = 1,    // Treppenstufen rechts
 isCenteredSteps = 2, // Treppenstufen mittig
 isStepsLeft = 3,     // Treppenstufen links
 isBinary = 4,        // binär
}
```

enumFillStyle

Definiert die verschiedenen Füllstile einer Hüll-bzw. Flutkurve.

```
enum enumFillStyle
{
 fsSolid = 0,      // ausgefüllt
 fsHorizontal = 1, // horizontal /* ----- */
 fsVertical = 2,   // vertikal   /* ||||| */
 fsFDiagonal = 3,  // diagonal   /* \\\\\ */
 fsBDiagonal = 4,  // diagonal   /* ///// */
 fsCross = 5,      // gekreuzt   /* +++++ */
 fsDiagCross = 6,  // X          /* xxxxx */
}
```

enumFloodStyle

Definiert die Art der Flutung.



```
enum enumFloodStyle
{
 fsNone = 0,        // kein Fluten
 fsYValue = 1,      // Fluten gegen einen Y-Wert
 fsCurve = 2,       // Fluten gegen eine Kurve
}
```

enumLineStyle

Definiert die verschiedenen Linienstile.

```
enum enumLineStyle
{
 lsSolid = 0,       // durchgehend          /* _____ */
 lsDash = 1,        // gestrichelt          /* _ _ _ */
 lsDot = 2,         // gepunktet            /* ..... */
 lsDashDot = 3,     // Strich-Punkt         /* ._._. */
 lsDashDotDot = 4,  // Strich-Punkt-Punkt   /* ///// */
 lsNull = 5,        // nicht sichtbar
}
```

enumMargins

Definiert die Randbereiche des *TrendViewers*.

```
enum enumMargins
{
 rmLeft = 0,        // links
 rmTop = 1,         // oben
 rmRight = 2,       // rechts
 rmBottom = 3,      // unten

}
```

enumPrintLegendStyles

Definiert die Identifikationsnummer zur Positionierung der Legende beim Ausdrucken der Trend-Komponente.

```
enum enumPrintLegendStyles
{
 plNoLegend = 0,    // keine Legende
 plLeft = 1,        // Legende links vom Trend
 plTop = 2,         // Legende an der Trend-Oberseite
 plRight = 3,       // Legende rechts vom Trend
 plBottom = 4,      // Legende an der Trend-Unterseite
}
```

enumRulerStyle

Definiert die Stile für die Hilfslineale.



```
enum enumRulerStyle
{
 rsNoRuler         = 0,// kein Lineal
 rsVerticalRuler   = 1,// vertikales Lineal
 rsHorizontalRuler = 2,// horizontales Lineal
}
```

enumStripChartsStyle

Definiert alle Stile für die Stripcharts.

```
enum enumStripChartsStyle
{
 scNoStripCharts   = 0,// keine Stripcharts
 scLeftSide        = 1,// Stripcharts auf der linken Seite
 scRigthSide       = 2,// Stripcharts auf der rechten Seite
 scBothSides       = 3,// Stripcharts auf beiden Seiten
}
```

enumScaleGroup

Definiert die möglichen Skalen-Gruppen, denen die Datenskalen in der Trend-Komponente zugeordnet werden können.

```
enum enumScaleGroup
{
 sgTop    = 0,      // Skala liegt an der Trend-Oberseite
 sgBottom = 1,      // Skala liegt an der Trend-Unterseite
 sgLeft   = 2,      // Skala liegt links
 sgRight  = 3,      // Skala liegt rechts
}
```

enumScaleTypes

Definiert die möglichen Skalen-Gruppen, denen die Datenskalen in der Trend-Komponente zugeordnet werden können.

```
enum enumScaleTypes
{
 stDecimal = 0,     // Dezimalskala
 stTime   = 1,      // Zeitskala (nur x-Skala)
 stDegree = 2,      // Winkelskala
 stLogarithmic = 3, // logarithmische Skala
}
```

enumTextAlign

Definiert die Textausrichtung in der Skalenlegende.



```
enum enumTextAlign
{
 taCenter = 0,      // mittig
 taLeft = 1,        // links
 taRight = 2,       // rechts
}
```

enumMouseCursor

Kenn-Nummer für die Maus-Cursor des Trends.

```
enum enumMouseCursor
{
 mcDefault = 0,     // Standard-Cursor(Skalen)
 mcTrans = 1,       // transparenter Cursor
 mcRulerVert = 2,   // Cursor für das vertikale Lineal
 mcRulerHoriz = 3,  // Cursor für das horizontale Lineal
 mcAuxline = 4,     // Cursor für die Hilfslinien
 mcCurve = 5,       // Cursor für eine Kurve
 mcCurvePoint = 6,  // Cursor für einen Kurvenpunkt
 mcScaleL = 7,      // linker Mausknopf-Cursor(Skalen)
 mcScaleR = 8,      // linker Mausknopf-Cursor(Skalen)
 mcScaleB = 9,      // mittlerer Mausknopf-Cursor(Skalen)
 mcSize = 10,       // Cursor für Größenänderung (Skalen)
 mcTrendL = 11,     // linker Mausknopf-Cursor(Trend)
 mcTrendR = 12,     // rechter Mausknopf-Cursor(Trend)
 mcTrendB = 13,     // mittlerer Mausknopf-Cursor(Trend)
}
```

enumTrendZoom

Definiert die Identifikationsnummer für das Zoomen im Trend-Fenster.

```
enum enumTrendZoom
{
 tzNoZoom = 0,      // kein Zoomen
 tzZoom = 1,        // Zoomen (verschieben,stauchen,
                       spreizen)
 tzLassoZoom = 2,   // Lassozoom
}
```

enumXScalePosRows

Definiert die Skalen-Zeilen innerhalb einer x-Skalen-Spalte.

```
enum enumXScalePosRows
{
 srXRow1  = 0,       // 1. Zeile
 srXRow2  = 1,       // 2. Zeile
}
```



enumXScalePosColumn

Definiert die Spalten innerhalb der x-Skalen-Gruppen.

```
enum enumXScalePosColumn
{
 scXColumn1= 0,     // 1. Spalte
 scXColumn2= 1,     // für spätere Implementierung)
}
```

enumXScales

Definiert die Bezeichner für die x-Skalen im Trend.

```
enum enumXScales
{
 dsXScale1= 0,      // 1. X-Skala
 dsXScale2= 1,      // 2. X-Skala
}
```

enumYScalePosRows

Definiert die Skalen-Zeilen innerhalb einer y-Skalen-Spalte.

```
enum enumYScalePosRows
{
 srYRow1  = 0,      // 1. Zeile
 srYRow2  = 1,      // 2. Zeile
 srYRow3  = 1,      // 3. Zeile
 srYRow4  = 1,      // 4. Zeile
 srYRow5  = 1,      // 5. Zeile
 srYRow6  = 1,      // 6. Zeile
 srYRow7  = 1,      // 7. Zeile
 srYRow8  = 1,      // 8. Zeile
 srYRow9  = 1,      // 9. Zeile
 srYRow10 = 1,      // 10. Zeile
}
```

enumYScalePosColumn

Definiert die Spalten innerhalb der y-Skalen-Gruppen.



```
enum enumYScalePosColumn
{
 scYColumn1= 0,     // 1. Spalte
 scYColumn2= 1,     // 2. Spalte
 scYColumn3= 1,     // 3. Spalte
 scYColumn4= 1,     // 4. Spalte
 scYColumn5= 1,     // 5. Spalte
 scYColumn6= 1,     // 6. Spalte
 scYColumn7= 1,     // 7. Spalte
 scYColumn8= 1,     // 8. Spalte
 scYColumn9= 1,     // 9. Spalte
 scYColumn10= 1,    // 10. Spalte
}
```

enumYScales

Definiert die Bezeichner für die y-Skalen im Trend.

```
enum enumYScales
{
 dsYScale1= 0,      // 1. Y-Skala
 dsYScale1= 1,      // 2. Y-Skala
 dsYScale1= 1,      // 3. Y-Skala
 dsYScale1= 1,      // 4. Y-Skala
 dsYScale1= 1,      // 5. Y-Skala
 dsYScale1= 1,      // 6. Y-Skala
 dsYScale1= 1,      // 7. Y-Skala
 dsYScale1= 1,      // 8. Y-Skala
 dsYScale1= 1,      // 9. Y-Skala
 dsYScale1= 1,      // 10. Y-Skala
}
```



5.5 Linealfenster (Professional Edition)

5.5.1 Eigenschaften

Allgemeine Eigenschaften des Linealfenster:

Dispatch	Koppelt das Fenster an den Trendviewer

Dispatch

`TrendViewer3.Net20.TrendViewer` Dispatch

Beschreibung

Mit dieser Eigenschaft wird das Linealfenster an den Trendviewer gekoppelt.

Wertebereich

`TrendViewer3.Net20.TrendViewer` TrendviewerCtrl

Programm-Beispiel

```
Tv3_LegendWnd1.Dispatch = m_Trend
```

5.5.2 Ereignisse

Die folgende tabellarische Darstellung enthält eine Übersicht über alle Ereignisse der Legenden-Komponente:

Event Groups

XScFormatText	wird verschickt, bevor der Werte-Text einer x-Skala ausgegeben wird
YScFormatText	wird verschickt, bevor der Werte-Text einer y-Skala ausgegeben wird
CurveFormatText	wird verschickt, bevor der Werte-Text einer Kurve ausgegeben wird

XScFormatText

XScFormatText (Short ScaleID, Double Value, ref String Text)

Beschreibung

Tritt auf, bevor der Werte-Text für eine x-Skala ausgegeben wird. Der Ausgabetext kann somit geändert oder formatiert werden.

Parameter

ScaleID	zugehörige SkalenID
Value	der refernezierte String als Wert
Text	Referenz zu einem String



Siehe auch

YScFormatText, CurveFormatText

YScFormatText

YScFormatText (Short ScaleID, Double Value, ref String Text)

Beschreibung

Tritt auf, bevor der Werte-Text für eine y-Skala ausgegeben wird. Der Ausgabetext kann somit geändert oder formatiert werden.

Parameter

ScaleID	zugehörige SkalenID
Value	der refernezierte String als Wert
Text	Referenz zu einem String

Siehe auch

XScFormatText, CurveFormatText

CurveFormatText

CurveFormatText (Short CurveID, Double Value, ref String Text)

Beschreibung

Tritt auf, bevor der Werte-Text für eine Kurve ausgegeben wird. Der Ausgabetext kann somit geändert oder formatiert werden.

Parameter

CurveID	zugehörige CurvenID
Value	der refernezierte String als Wert
Text	Referenz zu einem String

Siehe auch

XScFormatText, YScFormatText